



普通高等教育“十三五”规划教材

# 程序设计基础(C/C++语言)

© 邹启明 主编



中国工信出版集团



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

普通高等教育“十三五”规划教材

# 程序设计基础 (C/C++语言)

邹启明 主编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书重点介绍在 C 语言环境下编写程序的思路与方法, 主要讲述 C 语言的数据类型、运算规则, 顺序、选择、循环结构的程序设计, 数组、函数、指针、结构体和文件等内容, 在介绍基本概念、基本语法及常规算法的基础上, 强调模块化、规范化的程序设计。C++ 语言作为一种高效、实用的程序设计语言, 既可以进行过程化程序设计, 也可以进行面向对象程序设计, 为引导读者逐步体会面向对象程序设计的特点, 在有关章节中增加了 C++ 面向对象程序设计内容, 涉及类和对象、类的构造和析构以及输入流/输出流等内容。

本书可作为高等学校计算机及相关专业教材, 也可供从事计算机相关领域的科研人员自学参考。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

### 图书在版编目 (CIP) 数据

程序设计基础: C/C++ 语言 / 邹启明主编. —北京: 电子工业出版社, 2017.1

ISBN 978-7-121-30509-2

I. ①程… II. ①邹… 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 (2016) 第 287106 号

策划编辑: 冉 哲

责任编辑: 冉 哲

印 刷: 北京京科印刷有限公司

装 订: 北京京科印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 14.25 字数: 364.8 千字

版 次: 2017 年 1 月第 1 版

印 次: 2017 年 4 月第 2 次印刷

定 价: 35.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

本书咨询联系方式: [ran@phei.com.cn](mailto:ran@phei.com.cn)。

# 前 言

程序设计基础是高等学校理工类专业学生的编程入门基础课程，为引导学生有效学习本课程，本书以 C/C++ 语言作为基本工具，以程序设计的思想与方法作为核心内容，以动手编程实际问题能力的培养为最终目标。通过本书的学习，不仅使学生掌握程序设计语言本身的语法与结构，更重要的是逐步培养学生用计算机解决问题的思维、习惯与方法。

本书在对概念的讲解上注重强调基本语法和基本结构在编程中的作用及其所实现的功能，而不去罗列一些具体的语法细节和特例，从宏观上把握程序的结构。

本书的读者对象是没有编程基础的初学者，通过本书所能接触到的也只是一些相对简单的程序，但程序结构的设计和编程习惯的培养却正是从初学时开始形成的，因此本书所涉及的概念、算法、语法，以及例题的讲解都强调规范化、结构化，引导读者适当地模仿，从基本程序的学习开始就养成规范编程的习惯。

本书采用大量的图示说明，把复杂的概念、算法用图形的形式描述出来，使读者有一个形象、直观的认识。

本书共 8 章。第 1 章介绍 C/C++ 程序的基本要素和开发步骤；第 2 章介绍 C/C++ 语言的基本数据类型、运算符与表达式以及数据类型转换，简单输入/输出语句等；第 3 章介绍结构化程序的设计方法，以及与三种控制结构——顺序、选择和循环结构相关联的语法知识及其控制语句；第 4 章介绍一维数组和二维数组的定义与使用，数组名作为函数参数的应用，字符数组与字符串的应用；第 5 章介绍函数的定义与声明，函数的调用，以及函数的作用域等内容；第 6 章介绍指针与指针变量的概念，以及指针运算、指针数组及函数指针等，另外，对结构体、单向链表的各种操作也做了详细的说明；第 7 章介绍 C/C++ 文件的打开、关闭、读取及写入等操作；第 8 章介绍类和对象的定义方法、对象的初始化、构造函数和析构函数等。

在本书的编写过程中，编者参阅了许多参考书和有关资料，谨向这些书的作者表示衷心的感谢！

本书由邹启明（第 1 章、第 8 章及附录）、严颖敏（第 2 章）、高洪皓（第 3 章、第 4 章）、庄伟明（第 5 章）、朱弘飞（第 6 章）、王萍（第 7 章）等编写。在本书的编写过程中，陈章进、杨利明、高珏、宋兰华、马剑锋、单子鹏、余俊、马骄阳、陶媛、王文、张军英、钟宝燕等老师对本书的内容提出了很多宝贵意见，在此一并表示衷心的感谢。

由于时间仓促，作者水平有限，书中难免有错误之处，敬请读者批评指正。

编 者

# 目 录

第 1 章 程序设计基础	(1)
1.1 简单的 C/C++ 程序	(1)
1.1.1 输出“Hello,World!”	(1)
1.1.2 求解并输出 $n$ 的阶乘值	(2)
1.1.3 使用函数实现求解 $n$ 的阶乘值	(3)
1.2 算法	(4)
1.2.1 算法的概念及表示方法	(4)
1.2.2 算法设计的基本方法	(8)
1.3 程序与程序设计	(15)
1.3.1 程序与程序设计语言	(16)
1.3.2 程序设计语言处理过程	(17)
1.3.3 程序设计步骤	(20)
1.3.4 程序设计方法	(21)
1.3.5 程序设计规范	(23)
1.4 小结	(24)
第 2 章 C 语言基础知识	(25)
2.1 标识符	(25)
2.2 基本数据类型	(25)
2.2.1 整型常量与变量	(26)
2.2.2 浮点型常量与变量	(27)
2.2.3 字符常量与变量	(28)
2.2.4 字符串常量	(28)
2.2.5 符号常量	(29)
2.3 基本输入/输出函数	(29)
2.3.1 格式化输出函数	(29)
2.3.2 格式化输入函数	(32)
2.3.3 字符输出函数	(34)
2.3.4 字符输入函数	(35)
2.4 运算符与表达式	(35)
2.4.1 算术运算符与表达式	(35)
2.4.2 赋值运算符与表达式	(36)
2.4.3 关系运算符与表达式	(37)
2.4.4 逻辑运算符与表达式	(37)
2.4.5 条件运算符与表达式	(39)
2.4.6 位运算符与表达式	(39)



2.4.7	逗号运算符与逗号表达式	(41)
2.4.8	sizeof 运算符	(41)
2.4.9	类型转换	(41)
2.5	C++语言的输入/输出	(42)
2.6	小结	(43)
<b>第3章</b>	<b>程序控制结构</b>	(44)
3.1	顺序结构	(44)
3.2	选择结构	(45)
3.2.1	if 语句	(45)
3.2.2	switch 语句	(50)
3.3	循环结构	(54)
3.3.1	while 循环语句	(54)
3.3.2	do-while 循环语句	(55)
3.3.3	for 循环语句	(56)
3.3.4	break 语句与 continue 语句	(58)
3.3.5	循环结构的嵌套	(59)
3.3.6	典型例题	(62)
3.4	常用算法	(68)
3.4.1	顺推法	(68)
3.4.2	倒推法	(69)
3.4.3	迭代法	(72)
3.4.4	穷举法	(75)
3.5	小结	(75)
<b>第4章</b>	<b>数组</b>	(77)
4.1	一维数组	(77)
4.1.1	定义	(77)
4.1.2	存储	(78)
4.1.3	引用	(78)
4.1.4	初始化	(79)
4.1.5	典型例题	(80)
4.2	二维数组	(87)
4.2.1	定义	(87)
4.2.2	存储	(87)
4.2.3	引用	(88)
4.2.4	初始化	(88)
4.2.5	典型例题	(89)
4.3	字符数组	(92)
4.3.1	字符串的表示	(93)
4.3.2	字符串处理函数	(94)
4.3.3	字符串的检索、插入和删除	(96)

4.4	用 C++ 语言处理字符串	(102)
4.5	小结	(109)
<b>第 5 章</b>	<b>函数</b>	<b>(110)</b>
5.1	程序与函数	(110)
5.2	函数的定义	(113)
5.3	函数的调用	(114)
5.4	数组名作为函数的参数	(119)
5.5	函数的应用	(120)
5.6	函数的嵌套与递归	(123)
5.6.1	函数的嵌套调用	(123)
5.6.2	函数的递归调用	(124)
5.7	局部变量与全局变量	(127)
5.7.1	局部变量	(127)
5.7.2	全局变量	(128)
5.8	变量的存储方式	(131)
5.8.1	变量存储的内存空间分布	(131)
5.8.2	变量存储类别	(132)
5.9	小结	(133)
<b>第 6 章</b>	<b>指针与结构体</b>	<b>(134)</b>
6.1	指针	(134)
6.1.1	变量的内存地址	(134)
6.1.2	指针变量	(135)
6.1.3	指针与数组	(140)
6.1.4	指针与函数	(143)
6.1.5	指针与字符串	(149)
6.2	结构体	(153)
6.2.1	结构体类型与结构体变量	(154)
6.2.2	结构体数组	(157)
6.2.3	结构体指针	(159)
6.2.4	结构体与函数	(161)
6.3	用指针和结构体处理链表	(163)
6.3.1	链表的概念	(163)
6.3.2	建立链表	(164)
6.3.3	输出链表	(166)
6.3.4	插入结点的操作	(166)
6.3.5	删除结点的操作	(167)
6.3.6	用 C++ 语言实现链表	(168)
6.4	小结	(173)
<b>第 7 章</b>	<b>文件</b>	<b>(174)</b>
7.1	文件概述	(174)

7.1.1	流	(174)
7.1.2	文件的类型	(174)
7.1.3	缓冲文件系统	(175)
7.1.4	文件指针	(175)
7.2	文件的打开与关闭	(176)
7.2.1	打开文件	(176)
7.2.2	关闭文件	(178)
7.3	文件的读/写	(179)
7.3.1	读/写文本文件	(179)
7.3.2	读/写二进制文件	(185)
7.4	文件其他相关函数	(187)
7.4.1	文件定位函数	(187)
7.4.2	错误检测函数	(192)
7.4.3	文件管理函数	(193)
7.5	C++语言中的文件	(194)
7.5.1	文件的打开和关闭	(194)
7.5.2	文本文件的读/写	(195)
7.5.3	二进制文件的读/写	(196)
7.6	小结	(197)
<b>第 8 章</b>	<b>C++面向对象程序设计基础</b>	<b>(198)</b>
8.1	面向对象的基本概念	(198)
8.2	类和对象	(200)
8.2.1	从结构到类	(200)
8.2.2	类的定义	(201)
8.2.3	对象的定义	(203)
8.2.4	构造函数	(203)
8.2.5	析构函数	(205)
8.3	小结	(207)
<b>附录 A</b>	<b>常用 ASCII 字符表</b>	<b>(208)</b>
<b>附录 B</b>	<b>常用的 ANSI C 标准库函数</b>	<b>(210)</b>
<b>附录 C</b>	<b>C 语言运算符优先级与结合性</b>	<b>(218)</b>
	<b>参考文献</b>	<b>(220)</b>



# 第 1 章 程序设计基础

语言，是人与人进行交流沟通的工具。人与计算机通信也需要语言，为了使计算机进行各种工作，就需要有一套用以编写计算机程序的数字、字符和语法规则，由这些字符和语法规则组成计算机的各种指令（或各种语句），这些就是计算机所能接受的语言，称为计算机语言。

## 1.1 简单的 C/C++ 程序

C 语言是目前世界上广泛使用的高级语言，其结构紧凑、语言简洁，只有 32 个关键字，9 种控制语句；使用方便灵活，书写形式自由；数据类型完备，运算符丰富；允许直接访问物理地址，对硬件进行操作；生成目标代码质量高，程序执行效率高；可用于各种型号的计算机和操作系统。

C 语言是结构化和模块化的语言，它是面向过程的。在处理较小规模的程序时，程序设计者用 C 语言较为得心应手；但是当面对的问题比较复杂、程序规模比较大时，程序设计者往往感到力不从心。

C++ 语言的目的是在 C 语言的基础上实现面向对象编程的功能，从而实现程序代码更易维护、代码重用性更高、程序结构更清晰且具有 C 语言的高效性。在 C 语言的基础上新增了一些数据类型，如 Boolean（逻辑型）、Const（常量）等，并且使用内联函数、引用、重载、继承、虚函数等，实现面向对象的程序设计功能。

### 1.1.1 输出 “Hello, World!”

【例 1-1】 使用 C 语言编写程序输出 “Hello, World!”。

```
#include <stdio.h>

int main() /*主程序部分，main 为程序入口*/
{
    printf("Hello, World!\n"); //printf() 为输出函数
    return 0;
}
```

运行结果：

**Hello, World!**

程序说明：

① #include <stdio.h> 指示编译器在对程序进行预处理时，将文件 stdio.h (standard input and output, 标准输入/输出) 中的代码嵌入到程序中该指令所在的地方，这样该程序在正式编译时，就可以利用包含文件中的内容了。

② 每个 C 程序都必须有并且只有一个 main() 主函数，main 的名字也不可被更改。一个 C 程序可以包含若干个函数，总是从主函数处开始执行，main() 函数前面的 int 表示本函数返回值是“整型”的意思，也可用 void 表示没有返回值。

③ 花括号 {} 括起来的表示函数体，由语句组成。分号 “;” 是语句结束符，表示该语句结束。printf() 为输出函数，符号 “\n” 表示换行，即输出完内容之后，将光标移到下一行开头（最

左边)处。

④ 符号“/\*.....\*/”和“//”为注释语句,它只帮助阅读和理解程序,可以起到注释、标识、说明、指示等作用,以增加程序的可读性。这部分内容不参加程序的编译,对程序的功能无影响。“/\*.....\*/”可以跨行或跨段注释,“//”只注释本行。

**【例 1-2】** 使用 C++语言编写程序输出“Hello,World!”。

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello,World!"<<endl;
    return 0;
}
```

运行结果:

**Hello,World!**

程序说明:

① `iostream` 指标准输入/输出流; C++中为了避免名字定义冲突,特别引入了“名字空间”的定义,即 `namespace`,用于标识符的各种可见范围。C++标准程序库中的所有标识符都被定义于一个名为 `std` 的 `namespace` 中。

② `cin` 和 `cout` 是标准设备,用来实现数据的输入和输出。`cin` 一般代表键盘(输入),`cout` 一般代表显示器(输出)。

③ `endl` 类似于 C 语言中的 `\n`,作用是换行(输出)。

## 1.1.2 求解并输出 $n$ 的阶乘值

**【例 1-3】** 使用 C 语言编写一个循环程序,输入  $n$  值,输出  $n$  的阶乘 ( $n!$ ) 值。

分析:所谓  $n$  的阶乘就是从 1 到  $n$  的累积,所以可以通过一个 `for` 循环语句,从 1 到  $n$  依次求积即可。

```
#include <stdio.h>
int main()
{
    int i,n,fa=1;
    scanf("%d",&n);
    for(i=1;i<=n;i++) //for 循环语句,初值为 1,终值为 n,步长为 1
        fa=fa*i;
    printf("%d\n",fa);
    return 0;
}
```

运行结果(输入 5,输出 120):

**5**  
**120**

**【例 1-4】** 使用 C++语言编写一个循环程序,输入  $n$  值,输出  $n$  的阶乘 ( $n!$ ) 值。

```

#include <iostream>
using namespace std;
int main()
{
    int i,n,fa=1;
    cin>>n;
    for(i=1;i<=n;i++)
        fa=fa*i;
    cout<<fa<<endl;
    return 0;
}

```

运行结果（输入 6，输出 720）：

```

6
720

```

### 1.1.3 使用函数实现求解 $n$ 的阶乘值

**【例 1-5】** 使用 C 语言函数形式编写程序，输入  $n$  值，输出  $n$  的阶乘 ( $n!$ ) 值。

```

#include <stdio.h>
int factorial(int n)//定义求 n 的阶乘值的函数
{
    int i,fa=1;
    for(i=1;i<=n;i++)
        fa=fa*i;
    return fa; //返回函数值
}
int main()
{
    int n,f;
    scanf("%d",&n);
    f=factorial(n); //调用函数
    printf("%d\n",f);
    return 0;
}

```

运行结果（输入 5，输出 120）：

```

5
120

```

一个 C/C++ 程序可由一个主函数和若干个其他函数构成，由主函数调用其他函数，其他函数之间也可以互相调用。C/C++ 程序从 `main()` 函数处开始运行，当 `main()` 函数结束时，程序也就结束了。函数具有代码重用、提高编写效率和利于程序维护等诸多优点，读者可详细阅读第 5 章。

总结:

- ① C/C++程序是由一个或多个函数构成的，并且只能有一个主函数。
- ② 不管有多少个函数，程序执行都是从 `main()`函数开始的。在一个函数内，执行顺序是从上到下的。
- ③ 注释是从“//”开始的，具有增加可读性的作用。
- ④ 程序书写形式自由，一行内可以写多条语句，每条语句以“;”结束。
- ⑤ C语言区分大小写字母。

## 1.2 算法

算法 (Algorithm) 是指对解题方案的准确而完整的描述，是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制。算法是计算机科学最基本的概念，了解算法及其表示和设计方法是程序设计的基础和精髓。

### 1.2.1 算法的概念及表示方法

#### 1. 算法及其特性

##### (1) 什么是算法

算法就是一组有穷的规则，它规定了解决某个特定问题的一系列运算。通俗地说，为解决问题而采用的方法和步骤就是算法。

##### (2) 算法的特性

① 确定性 (Definiteness)。算法的每个步骤必须要有确切的含义，每个操作都应当是清晰的、无二义性的。

② 有穷性 (Finiteness)。一个算法应包含有限的操作步骤且在有限的时间 (人们可以接受的) 内能够执行完毕。

③ 有效性 (Effectiveness)。算法中的每个步骤都应当能有效地执行，并得到确定的结果。

④ 有零个或多个输入 (Input)。在算法执行的过程中需要从外界取得必要的信息，并以此为基础解决某个特定问题。

⑤ 有一个或多个输出 (Output)。设计算法的目的就是要解决问题，算法的计算结果就是输出。没有输出的算法是没有意义的。输出与输入有着特定的关系，通常，输入不同，会产生不同的输出结果。

##### (3) 算法的分类

根据待解决问题的形式模型和求解要求，算法分为数值和非数值两大类。

① 数值运算算法：是以数学方式表示的问题求数值解的方法。例如，代数方程计算、线性方程组求解、矩阵计算、数值积分、微分方程求解等。通常，数值运算有现成的模型，这方面的现有算法比较成熟。

② 非数值运算算法：通常为求非数值解的方法。例如，排序、查找、表格处理、文字处理、人事管理、车辆调度等。

#### 2. 算法的表示方法

设计出一个算法后，为了存档，以便将来算法的维护或优化，或者为了与他人交流，让他人能够看懂、理解算法，需要使用一定的方法来描述、表示算法。算法的表示方法很多，

常用的有自然语言、流程图和伪代码等。我们以求解  $\text{sum}=1+2+3+4+5+\dots+(n-1)+n$  为例，使用这三种不同的表示方法去描述解决问题的过程。

### (1) 自然语言 (Natural Language)

用人们日常生活中使用的语言，如中文、英文、法文等来描述算法。中文描述上述从 1 开始的连续  $n$  个自然数求和问题的算法如下：

- S1 确定一个  $n$  的值；
- S2 假设等号右边的算式项中的初始值  $i$  为 1；
- S3 假设  $\text{sum}$  的初始值为 0；
- S4 如果  $i \leq n$ ，则执行 S5，否则转出执行 S8；
- S5 计算  $\text{sum}$  加上  $i$  的值后，重新赋值给  $\text{sum}$ ；
- S6 计算  $i$  加 1，然后将值重新赋值给  $i$ ；
- S7 转去执行 S4；
- S8 输出  $\text{sum}$  的值，算法结束。

使用自然语言描述算法的优点是通俗易懂，没有学过算法相关知识的人也能够看懂算法的执行过程。但是，自然语言本身所固有的不严密性使得这种描述方法存在以下缺陷：

- ① 文字冗长，容易产生歧义性；
- ② 难以描述算法中的分支和循环等结构，不够方便、直观。

### (2) 流程图 (Flow Chart)

流程图是最常见的算法图形化表达，有传统流程图和 N-S 流程图两种形式。传统流程图使用美国国家标准化学会 (American National Standards Institute, ANSI) 规定的一些图框、线条来形象、直观地描述算法处理过程。常见的流程图符号如表 1-1 所示。

表 1-1 常见流程图符号

符 号	名 称	作 用
	开始符、结束符	表示算法的开始和结束符号
	输入框、输出框	在算法过程中，表示从外部获取的信息（输入），然后将处理过的信息输出
	处理框	在算法过程中，表示需要处理的内容，只有一个入口和一个出口
	判断框	表示算法过程中的分支结构，菱形框的 4 个顶点，通常用上面的顶点表示入口，根据需要用其余的顶点表示出口
	流程线	表示算法过程中流程的方向

使用流程图描述从 1 开始的连续  $n$  个自然数求和的算法如图 1-1 所示。

在使用过程中，人们发现流程线并不一定是必需的。随着结构化程序设计方法的出现，1973 年美国学者 I.Nassi 和 B.Shneiderman 提出了一种新的流程图形式，这种流程图完全去掉了流程线，算法的每步都用一个矩形框来描述，把一个个矩形框按执行的顺序连接起来，就是一个完整的算法描述。这种流程图用两位学者名字的第一个字母来命名，称为 N-S 流程图。

为了提高算法的质量，便于阅读理解，应限制流程的随意转向。为了达到这个目的，人们规定了三种基本结构，由这些基本机构按一定规律组成一个算法结构。

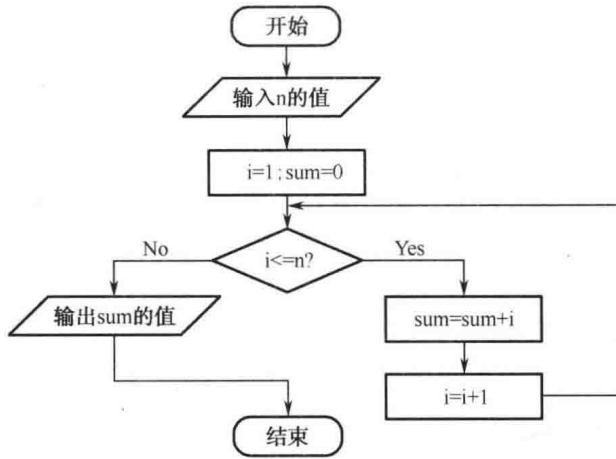


图 1-1 计算  $1+2+3+\dots+n$  的流程图

① 顺序结构：这是最简单的一种基本结构，各操作是按先后顺序执行的，如图 1-2 所示，图中操作 A 和操作 B 按照出现的先后顺序依次执行。

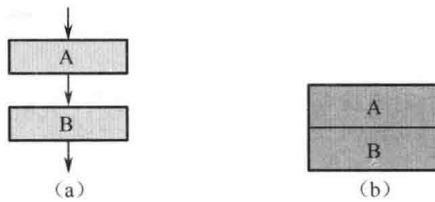


图 1-2 顺序结构

② 选择结构：又称分支结构，根据是否满足给定条件从两组操作中选择执行一种操作，某部分的操作可以为空操作。如图 1-3 所示，如果条件 P 成立则执行处理框 A，否则执行处理框 B。

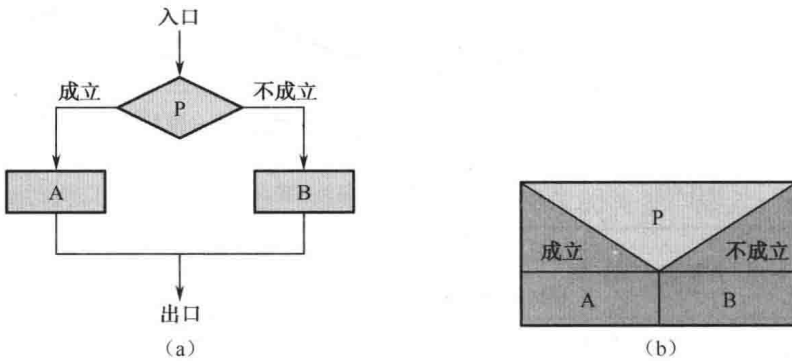


图 1-3 选择结构

③ 循环结构：又称重复结构，即在一定条件下，反复执行某部分的操作。循环结构又分为当型和直到型两种类型。

当型循环结构如图 1-4 所示，当条件 P 成立时，执行处理框 A，执行完处理框 A 后，再判断条件 P 是否成立，如果条件 P 成立，则再次执行处理框 A，如此反复，直至条件 P 不成立才结束循环。

直到型循环结构如图 1-5 所示，先执行处理框 A，再判断条件 P 是否成立，如果条件 P 不成立，则再次执行处理框 A，如此反复，直至条件 P 成立才结束循环。



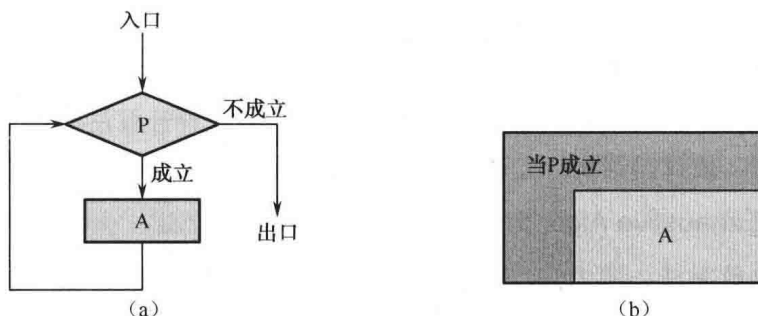


图 1-4 当型循环结构

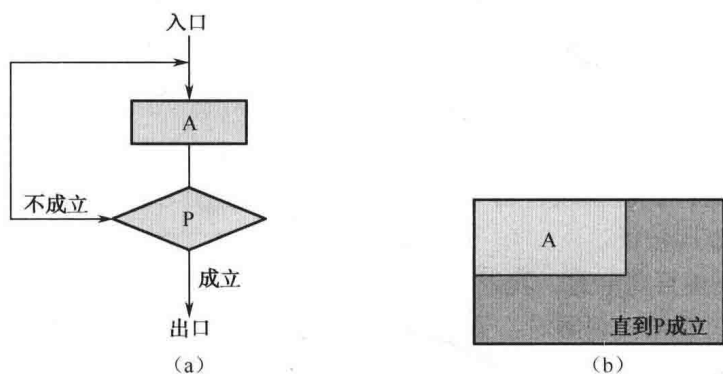


图 1-5 直到型循环结构

以上三种基本结构具有以下特点：

- ① 只有一个入口和一个出口；
- ② 结构内的每一部分都有机会被执行到；
- ③ 结构内不存在“死循环”（无终止的循环）。

### (3) 伪代码 (Pseudocode)

伪代码是一种用来书写程序或描述算法时使用的非正式、透明的表述方法。它并非是一种编程语言，这种方法针对的是一台虚拟的计算机。伪代码通常采用自然语言、数学公式和符号来描述算法的操作步骤，同时采用计算机高级语言（如 C、Pascal、VB、C++、Java 等）的控制结构来描述算法步骤的执行。

使用伪代码描述从 1 开始的连续  $n$  个自然数求和的算法如下：

- S1 算法开始；
- S2 输入  $n$  的值；
- S3  $i \leftarrow 1$ ； //为变量  $i$  赋初值
- S4  $sum \leftarrow 0$ ； //为变量  $sum$  赋初值
- S5 do while  $i \leq n$  //当变量  $i \leq n$  时，执行下面的循环体语句
- S6 {  $sum \leftarrow sum + i$ ；
- S7  $i \leftarrow i + 1$ ； }
- S8 输出  $sum$  的值；
- S9 算法结束。

## 1.2.2 算法设计的基本方法

算法设计的任务是对各类问题设计出良好的算法及研究设计算法的规律和方法。针对一个给定的实际问题，要找出确实行之有效的算法，就需要掌握设计的策略和基本方法。

### 1. 穷举法 (Exhaustive Algorithm)

穷举法也称为枚举法、蛮力法，是一种简单、直接解决问题的方法。使用穷举法解决问题的基本思路是：依次穷举问题所有可能的解，按照问题给定的约束条件进行筛选，如果满足约束条件，则得到一组解，否则不是问题的解。将这个过程不断地进行下去，最终得到问题的所有解。

要使用穷举法解决实际问题，应当满足以下两个条件：

- ① 能够预先确定解的范围并能以合适的方法列举；
- ② 能够对问题的约束条件进行精确描述。

穷举法的优点是：比较直观，易于理解，算法的正确性比较容易证明；缺点是：需要列举许多种状态，效率比较低。

**【例 1-6】**百钱买百鸡问题。我国古代数学家张丘建在《算经》中出了一道题：“鸡翁一，值钱五；鸡母一，值钱三；鸡雏三，值钱一。百钱买百鸡，问鸡翁、母、雏各几何？”。意思是：某个人有 100 钱，打算买 100 只鸡。公鸡 5 钱一只，母鸡 3 钱一只，小鸡 1 钱 3 只。请编写一个算法，算出如何能刚好用 100 钱买 100 只鸡？

此题可用穷举法来解，以三种鸡的个数为穷举对象（分别设为  $x$ ,  $y$  和  $z$ ），以三种鸡的总数和买鸡用去的钱的总数为判定条件，穷举各种鸡的个数。按题意列出方程组如下：

$$x+y+z=100 \quad (1)$$

$$5x+3y+z/3=100 \quad (2)$$

上述方程组有三个未知数，两个方程式，所以  $x, y, z$  有多组解。我们可以用“穷举法”把  $x, y, z$  可能满足要求的组合列举出来，并判断是否符合要求，最后输出符合要求的组合。

假定  $x, y$  的值已知，那么由方程 (1) 可求得  $z$  值，而  $x, y$  可能的取值都在  $0 \sim 100$  之间，所以可以用二重循环来组合它们，每个  $x, y$  的组合都得到一个相应的  $z$  值，即  $z=100-x-y$ 。若  $x, y, z$  满足方程式 (2)，则输出  $x, y, z$ ，否则不输出。

该算法的流程如图 1-6 所示。

### 2. 递推法 (Recurrence)

递推法是一种重要的算法设计思想。一般从已知的初始条件出发，依据某种递推关系，逐次推出所要求的各中间结果及最后结果。其中，初始条件可能由问题本身给定，也可能是通过对问题的分析与化简后确定。在实际应用中，题目很少会直接给出递推关系式，而是需要通过分析各种状态，找出递推关系式，这也是应用递推法解决问题的难点所在。递推算法可分为顺推法和逆推法两种。

(1) 顺推法。从已知条件出发，逐步推算出要解决问题的方法。例如，斐波那契 (Fibonacci) 数列就可以通过顺推法不断推算出新的数据。

(2) 逆推法。也称为倒推法，是顺推法的逆过程，该方法从已知的结果出发，用迭代表达式逐步推算出问题开始的条件。

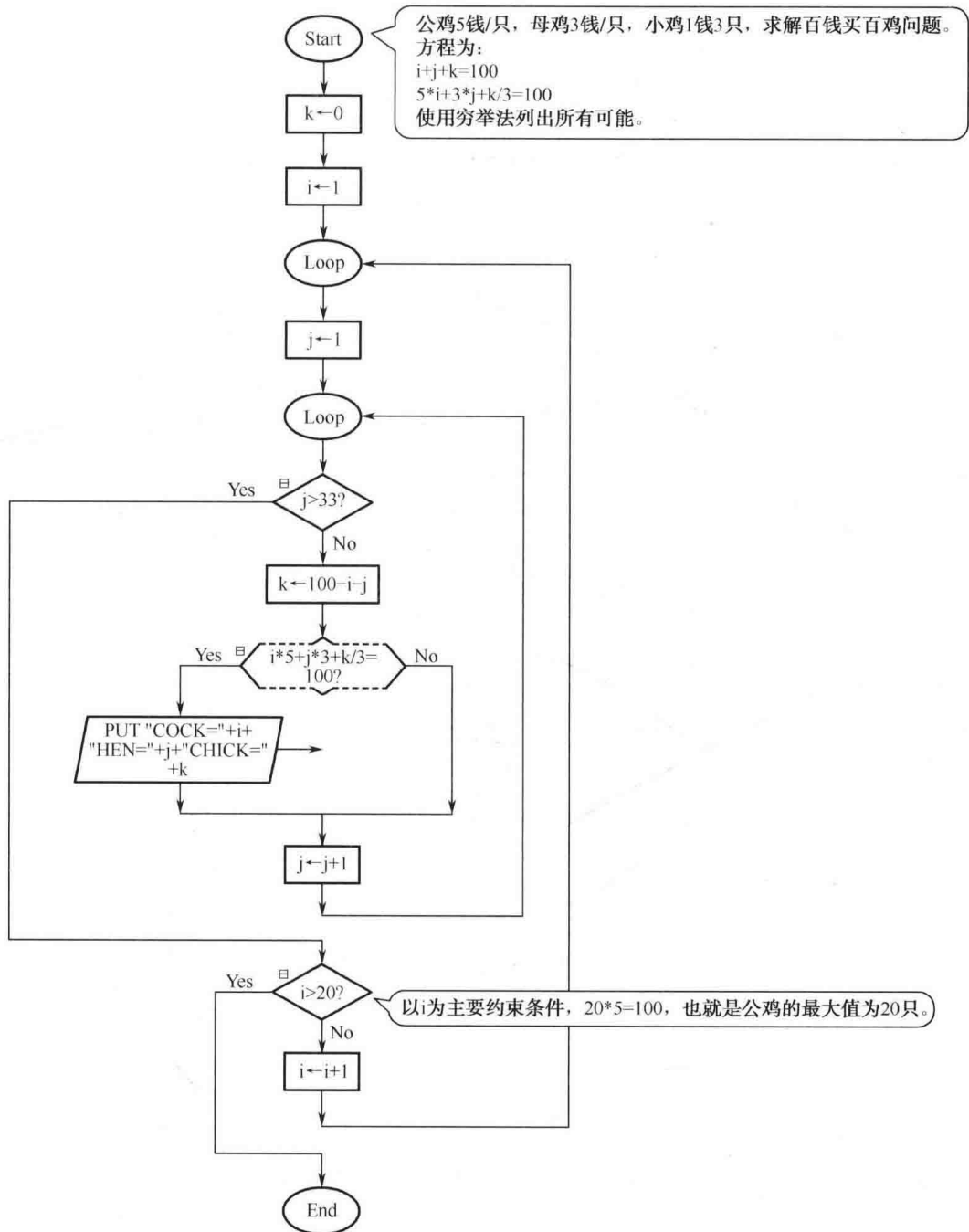


图 1-6 百钱买百鸡问题的流程图

**【例 1-7】** 使用顺推法解决 Fibonacci 数列问题。

Fibonacci 数列指的是这样一个数列：1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...这个数列从第 3 项开始，每项都等于前两项之和。使用数学公式表示如下：

$$\begin{cases} f_1 = 1 & (n = 1) \\ f_2 = 1 & (n = 2) \\ f_n = f_{n-1} + f_{n-2} & (n \geq 3) \end{cases}$$

从以上的分析可知，Fibonacci 数列可使用递推算算法来计算求得，图 1-7 就是使用顺推法解决 Fibonacci 数列前 12 项问题的流程图。