

福建省精品在线开放课程配套教材

■ 吴小菁 陈慧 杨玮 高建清 唐磊 / 编著

C语言程序设计 案例教程

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

C 语言程序设计案例教程

吴小菁 陈 慧 杨 玮 编著
高建清 唐 磊

 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 提 要

本书作为 C 语言程序设计的基础教材，共分为 9 章，主要包括 C 语言概述、C 语言程序设计的初步知识、顺序结构程序设计、选择结构程序设计、循环结构程序设计、函数、数组、编译预处理、指针等。本书在加强 C 语言基本知识训练的同时，注重对编程能力的培养，融合省级精品在线课程资源，结合二维码实现典型案例视频的立体呈现，通过精心设计的案例分析、深入浅出的解题技巧总结等，帮助学生更直观地理解知识点，掌握相应的操作技巧，从而极大地激发学生的学习兴趣。

本书具有通用性，既可作为普通高等院校的“C 语言程序设计”课程教材，又可供社会各类计算机应用人员与参加各类计算机等级考试的人员参考。

版权专有 侵权必究

图书在版编目 (CIP) 数据

C 语言程序设计案例教程 / 吴小菁等编著. —北京: 北京理工大学出版社, 2019. 7
(2019. 8 重印)

ISBN 978 - 7 - 5682 - 7299 - 5

I. ①C… II. ①吴… III. ①C 语言 - 程序设计 - 教材 IV. ①TP312. 8

中国版本图书馆 CIP 数据核字 (2019) 第 151391 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 三河市天利华印刷装订有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 11

字 数 / 260 千字

版 次 / 2019 年 7 月第 1 版 2019 年 8 月第 2 次印刷

定 价 / 36.00 元

责任编辑 / 梁铜华

文案编辑 / 曾 仙

责任校对 / 周瑞红

责任印制 / 李志强

图书出现印装质量问题，请拨打售后服务热线，本社负责调换

前 言

教育部在《教育信息化“十三五”规划》的文件中，明确了“人人皆学、处处能学、时时可学”的发展要求，移动互联网教育成为了首选。伴随着在线教育的发展，网络精品课程层出不穷，在线课程与线下课堂相结合的教学方式，便于学生自主学习，能促进师生互动，从而提高教学质量。但是，学生在学习过程中，如果对某个知识点有疑惑，往往希望能够得到快速解答，而网络精品课程虽内容丰富、课时量大，但需要花费大量时间去寻找对应的知识点。

针对以上情况，我们推出了这本基于典型案例、借助手机扫描二维码以获取教学资源的案例教材，使学生能够实时观看专题讲解，快速而便捷地浏览课程的图文信息。同时，本书还配套有精心设计的习题，便于学生课后进行自主练习，从而巩固知识点，提高学习效率。

本书作为C语言程序设计的基础教材，共分为9章，主要包括C语言概述、C语言程序设计的初步知识、顺序结构程序设计、选择结构程序设计、循环结构程序设计、函数、数组、编译预处理、指针。每一章都从知识梳理入手，给出若干典型案例；每个典型案例都采取任务驱动的教学方式讲解，并附有相应的二维码；典型案例之后是对本章知识点的小结；最后是配套习题。全书体现了C语言的基础知识与应用要求，通过实例来进一步提高学生的程序设计能力。本书以案例为主线，内容具有通用性，注重实践、应用面广，既可以作为普通高等院校的“C语言程序设计”课程教材，又可供社会各类计算机应用人员与参加各类计算机等级考试的人员参考。

本书的内容和视频由福建江夏学院的吴小菁、陈慧、杨玮、高建清和福建师范大学的唐磊共同编写与制作。其中，第1、2、9章由吴小菁编写，第3、7章由陈慧编写，第4、6章由杨玮编写，第5、8章由高建清编写；案例视频由吴小菁和唐磊制作，由吴小菁完成视频剪辑；全书由吴小菁统稿。

限于笔者的学识水平，书中难免存在疏漏和不妥之处，敬请广大读者批评指正。

编 者
2019年6月

第 1 章 C 语言概述	(1)
1.1 知识梳理	(1)
1.1.1 C 语言的发展及特点	(1)
1.1.2 C 语言程序的格式和构成	(2)
1.1.3 C 语言程序的编译过程	(4)
1.1.4 算法	(7)
1.2 典型案例	(9)
1.2.1 案例 1: Visual C++ 6.0 编译环境的应用	(9)
1.2.2 案例 2: Visual C++ 2010 编译环境的应用	(9)
1.3 本章小结	(9)
习题	(10)
第 2 章 C 语言程序设计的初步知识	(12)
2.1 知识梳理	(12)
2.1.1 C 语言的数据类型	(12)
2.1.2 标识符、常量和变量	(13)
2.1.3 整型数据	(15)
2.1.4 实型数据	(16)
2.1.5 字符型数据	(17)
2.1.6 算术运算	(18)
2.1.7 赋值运算	(19)
2.1.8 特殊运算	(20)
2.1.9 类型转换运算	(21)
2.2 典型案例	(21)
2.2.1 案例 1: 编程求圆的周长	(21)
2.2.2 案例 2: 计算表达式的值	(22)
2.2.3 案例 3: 编程实现时间换算	(22)



2.3 本章小结	(23)
习题	(25)
第3章 顺序结构程序设计	(27)
3.1 知识梳理	(27)
3.1.1 C语言的语句	(27)
3.1.2 数据的输出	(28)
3.1.3 数据的输入	(30)
3.2 典型案例	(32)
3.2.1 案例1: 赋值表达式与赋值语句	(32)
3.2.2 案例2: printf函数的应用	(33)
3.2.3 案例3: scanf函数的应用	(33)
3.2.4 案例4: 数据输入输出的综合应用	(33)
3.3 本章小结	(34)
习题	(36)
第4章 选择结构程序设计	(39)
4.1 知识梳理	(39)
4.1.1 C语言的逻辑值	(39)
4.1.2 关系运算	(39)
4.1.3 逻辑运算	(40)
4.1.4 if语句	(40)
4.1.5 switch语句	(41)
4.2 典型案例	(42)
4.2.1 案例1: 使用流程图描述算法	(42)
4.2.2 案例2: 计算表达式的值	(42)
4.2.3 案例3: 编程实现两个数的排序	(43)
4.2.4 案例4: 编程实现奇偶性的判断	(44)
4.2.5 案例5: 编程求两数的较大值	(44)
4.2.6 案例6: 编程实现成绩级别的判断	(45)
4.2.7 案例7: 编程求分段函数	(45)
4.2.8 案例8: switch语句的应用	(47)
4.3 本章小结	(48)
习题	(49)
第5章 循环结构程序设计	(53)
5.1 知识梳理	(53)
5.1.1 while语句	(53)
5.1.2 do...while语句	(54)



5.1.3	for 语句	(54)
5.1.4	break 和 continue 语句	(55)
5.1.5	goto 语句	(56)
5.1.6	循环结构的嵌套	(56)
5.2	典型案例	(56)
5.2.1	案例 1: 使用流程图描述算法	(56)
5.2.2	案例 2: 编程求 1~100 的累加和	(57)
5.2.3	案例 3: 编程求 $1^2 \sim n^2$ 的累加和	(58)
5.2.4	案例 4: 编程求 π 的近似值	(59)
5.2.5	案例 5: 编程实现固定行的输出	(60)
5.2.6	案例 6: 编程求 1~10 累积	(61)
5.2.7	案例 7: 编程求 1~10 000 奇数的累加和	(62)
5.2.8	案例 8: 编程求斐波那契数列项	(63)
5.2.9	案例 9: 编程实现一行字符的输入输出	(63)
5.2.10	案例 10: 编程实现矩阵的输出	(65)
5.2.11	案例 11: 编程实现图形的输出	(65)
5.2.12	案例 12: 编程输出 2~100 的素数	(66)
5.3	本章小结	(68)
	习题	(69)
第 6 章	函数	(75)
6.1	知识梳理	(75)
6.1.1	库函数	(75)
6.1.2	函数的定义	(76)
6.1.3	函数的返回值	(77)
6.1.4	函数的声明	(78)
6.1.5	函数的调用	(78)
6.1.6	函数的参数传递方式	(79)
6.1.7	函数的嵌套调用	(79)
6.1.8	函数的递归调用	(79)
6.1.9	变量的作用域和存储类型	(80)
6.1.10	函数的作用范围	(81)
6.2	典型案例	(82)
6.2.1	案例 1: 函数定义	(82)
6.2.2	案例 2: 无返回值的函数调用	(82)
6.2.3	案例 3: 有返回值的函数调用	(83)
6.2.4	案例 4: 阅读函数调用程序, 写运行结果	(84)
6.2.5	案例 5: 函数参数的值传递	(85)
6.2.6	案例 6: 函数实现素数的判断	(85)



6.2.7	案例7: 实现累加计算的函数	(86)
6.2.8	案例8: 阅读函数的嵌套调用程序, 写运行结果	(86)
6.2.9	案例9: 编写递归函数	(87)
6.3	本章小结	(89)
	习题	(90)
第7章	数组	(98)
7.1	知识梳理	(98)
7.1.1	一维数组	(98)
7.1.2	二维数组	(101)
7.1.3	字符数组	(103)
7.1.4	数组与函数	(105)
7.2	典型案例	(106)
7.2.1	案例1: 使用数组存放成绩	(106)
7.2.2	案例2: 编程实现一维数组的赋值	(107)
7.2.3	案例3: 编程实现一维数组的输入	(107)
7.2.4	案例4: 编程求数组的平均值	(108)
7.2.5	案例5: 编程实现二维数组的赋值	(109)
7.2.6	案例6: 编程求二维数组元素的累加和	(109)
7.2.7	案例7: 编程求二维数组主对角线元素的累加和	(110)
7.2.8	案例8: 运用函数改变数组元素	(111)
7.2.9	案例9: 运用函数求数组的平均值	(112)
7.2.10	案例10: 字符数组的处理	(113)
7.3	本章小结	(114)
	习题	(116)
第8章	编译预处理	(123)
8.1	知识梳理	(123)
8.1.1	宏定义	(123)
8.1.2	文件包含	(124)
8.2	典型案例	(125)
8.2.1	案例1: 带参宏的应用	(125)
8.3	本章小结	(126)
	习题	(127)
第9章	指针	(129)
9.1	知识梳理	(129)
9.1.1	指针概述	(129)
9.1.2	指针变量	(130)



9.1.3	指针与数组	(133)
9.1.4	指针与字符串	(136)
9.1.5	指针数组	(136)
9.1.6	指针与函数	(138)
9.1.7	指向指针的指针	(139)
9.2	典型案例	(140)
9.2.1	案例 1: 间接访问运算符的应用	(140)
9.2.2	案例 2: 指针与数组的应用	(141)
9.2.3	案例 3: 指针与字符串的应用	(141)
9.2.4	案例 4: 利用函数交换数据	(142)
9.2.5	案例 5: 利用指针传递数据	(143)
9.2.6	案例 6: 指针与函数的综合应用	(144)
9.2.7	案例 7: 二级指针的应用	(145)
9.3	本章小结	(146)
	习题	(147)
附录	(154)
附录 1	C 语言的关键字	(154)
附录 2	运算符的优先级和结合性	(155)
附录 3	常用字符与 ASCII 码对照表	(156)
附录 4	库函数	(157)
附录 5	习题参考答案	(162)
参考文献	(166)

第 1 章

C 语言概述

随着人工智能、物联网、大数据、云计算、机器人、虚拟现实等科学技术的蓬勃发展，信息技术的应用已经并正在继续改变着人们的工作和生活，而这些新技术和应用的核心是程序。程序是用某种程序设计语言编写、指示计算机完成特定功能的指令序列的集合。人人都应该了解程序、懂程序、会编程序，编程可以改变人们的思维方式，教会人们在这个时代应如何思考。学习编程需要选择一种合适的程序设计语言。C 语言既具有高级语言的面向过程的特点，又具有汇编语言的面向底层的特点。C 语言是目前最受欢迎、应用最广的高级语言之一，因此通常选择 C 语言作为首选的程序设计入门语言。

1.1 知识梳理

C 语言是一种通用的、面向过程的编程语言。C 语言广泛应用于系统与应用程序的开发，具有高效、灵活、功能丰富、表达力强和较高的可移植性等特点。本章将介绍 C 语言程序的基本结构、程序中的基本要求。

1.1.1 C 语言的发展及特点

1. 计算机语言的发展

计算机语言经历了三代：第一代是机器语言，第二代是汇编语言，第三代是高级语言。

1) 机器语言

对计算机本身来说，它并不能直接识别由高级语言编写的程序，它只能接收和处理由数字 0 和 1 组成的二进制指令。这种指令称为机器语言。

2) 汇编语言

为了便于编程，以及解决更加复杂的问题。程序员开始改进机器语言，使用英文缩写的



助记符来表示基本的计算机操作。这些助记符构成了汇编语言。目前，汇编语言仍然应用于工业电子等编程领域。

3) 高级语言

汇编语言虽然能编写高效率的程序，但是学习和使用汇编语言都不易，并且对于解决复杂的问题，汇编语言显得力不从心。于是，出现了高级语言。高级语言允许程序员使用接近日常英语的指令来编写程序。高级语言接近人的思维，通俗易懂，编程门槛和难度大大降低，如 C、C++、Java、Python 等编程语言都是高级语言。

2. C 语言的发展

1972 年，美国贝尔实验室的 Dennis M. Ritchie 在 B 语言的基础上设计并实现了 C 语言。

1978 年，Brain W. Kernighan 和 Dennis M. Ritchie 合著了影响深远的名著 *The C Programming Language* (《C 程序设计语言》)，成为以后 C 语言版本的基础，被称为旧标准 C 语言。

1983 年，美国国家标准局 (ANSI) 制定了新的 C 语言标准，称为 ANSI C。此后，陆续出现的各种 C 语言版本都是与之兼容的版本。

C 语言的设计影响了很多后来的编程语言，如 C++、Objective-C、Java、C#、Python 等。

3. C 语言的特点

C 语言的主要特点如下：

- (1) 具有结构化的控制语句及模块化结构。
- (2) 语言简洁，结构紧凑，使用方便、灵活。
- (3) 运算丰富，数据处理能力强。
- (4) 可以直接访问物理地址，实现对硬件和底层系统软件的访问。
- (5) 生成的代码质量高。
- (6) 可移植性好。

C 语言还有其他优点，如具有强大的图形功能、支持多种显示器和驱动器等。C 语言已成为一种通用程序设计语言。

1.1.2 C 语言程序的格式和构成

1. C 语言程序的格式

C 语言程序的书写格式很自由，既可以在一行写多条语句，也可以将一条语句分成多行写。C 语言程序的书写格式如下：

- (1) 在每条语句和数据定义的结尾，必须有一个分号。
- (2) 严格区分大小写字母。

(3) C 语言本身没有输入输出语句。输入输出操作是由 `scanf` 和 `printf` 等库函数来完成的。使用输入输出库函数之前，必须书写如下编译预处理命令行：

```
#include <stdio.h >
```

- (4) 编译预处理命令行前应添加 `#`，该命令行的结尾不能添加分号。

(5) 可以用“// ...”对程序的任何部分作行注释,也可以用“/* ... */”对程序进行块注释。

例如,编写一个C语言程序,输出“hello world!”。代码如下:

```
#include <stdio.h>
main()
{
    /* 下面语句要输出 hello world! */
    printf("hello world!\n");    //输出语句
}
```

2. C语言程序的构成

C语言程序的基本结构是函数,若干条语句构成一个函数,一个或多个函数构成一个程序。

例如,求两个数中的较大值。代码如下:

```
#include <stdio.h>    /* 编译预处理命令行 */
#include <stdlib.h>
int max_two(int x,int y)    //求两个数中较大值的函数
{
    int z;
    if(x>y)
        z=x;
    else
        z=y;
    return z;
}
main()    //主函数(入口函数)
{
    int a,b,c;
    scanf("%d%d",&a,&b);
    c=max_two(a,b);    //调用 max_two 函数
    printf("max = %d\n",c);
}
```

C语言中的函数分为两类:一类是库函数,可以直接在程序中使用(如printf和scanf等);另一类是用户自己定义的函数(如main和max_two等),这部分函数必须由用户自己编写代码。

每个函数均由函数首部与函数体两部分组成。函数首部通常由函数类型、函数名及函数参数组成;函数体由若干条语句组成。它们的基本组成如图1-1所示。

C语言程序的构成说明如下:

- (1) C语言支持顺序、选择、循环结构,所以C语言的程序是一种结构化程序。
- (2) C语言提供的“函数”实现了模块化结构。

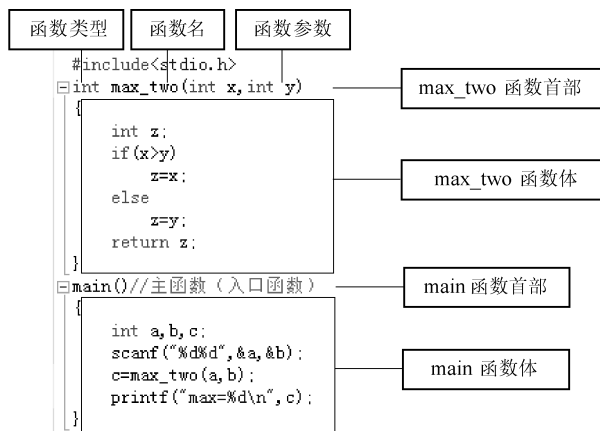


图 1-1 函数的基本组成

(3) C 程序可以包含任意多个不同名的函数，但必须有一个主函数，而且也只能有一个主函数。

(4) 主函数是入口函数，C 程序总是从主函数开始执行。

1.1.3 C 语言程序的编译过程

1. C 语言程序的编译和运行

C 语言是一种编译型的程序设计语言。一个 C 语言程序需要经过编辑、编译、连接和运行四个步骤，才能得到程序的运行结果。这四个步骤也称为 C 语言程序的调试和运行。

C 语言程序编辑后，利用 C 语言的编译程序对其进行编译，生成二进制代码表示的目标程序（扩展名为 .obj），利用 C 语言的连接程序把目标程序与 C 语言提供的各种库函数连接后生成可执行文件（扩展名为 .exe）。C 语言程序的调试、运行过程如图 1-2 所示。

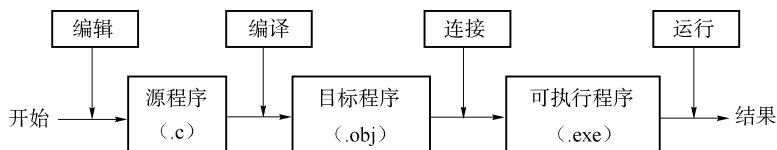


图 1-2 C 语言程序的调试、运行过程

2. C 语言程序的编译环境

目前所使用的大多数 C 语言编译系统都是集成环境。计算机端常用的有 Visual C++ 6.0、Visual C++ 2010 等，手机端常用的有 C4droid 与 C 语言编译器等。

下面简单介绍 Visual C++ 6.0、Visual C++ 2010 的编译环境。

1) Visual C++ 6.0 的编译环境

(1) 新建文件。打开 Visual C++ 6.0 程序，在文件菜单栏选择“新建”命令，弹出“新

建”对话框。在左侧的“文件”选项卡中单击“C++ Source File”图标，选择该文件类型，在右侧的“文件名”文本框中输入文件名称，然后通过“位置”的“浏览”按钮选择存储路径。最后，单击“确定”按钮，完成文件的新建。在 Visual C++ 6.0 编译环境中创建 C 语言源程序的“新建”对话框如图 1-3 所示。

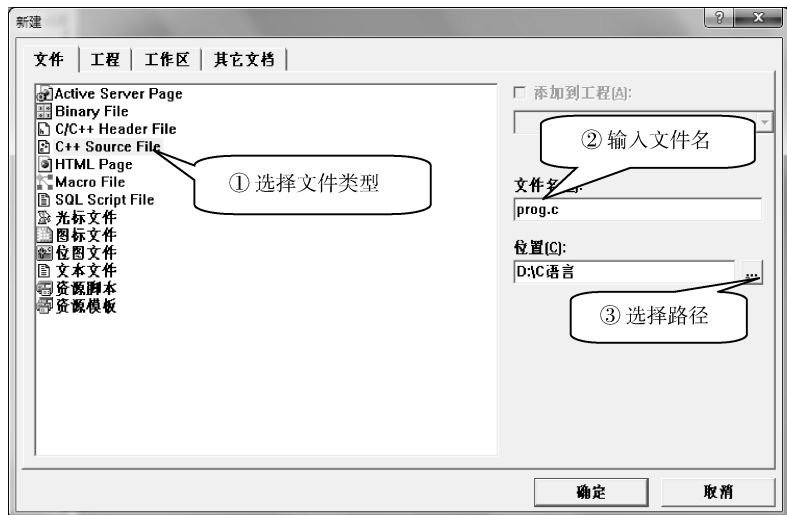


图 1-3 “新建”对话框 (Visual C++ 6.0)

(2) 调试和运行。在 C 语言编辑窗口输入代码后，先单击“编译”按钮，然后单击“连接”按钮，最后单击“运行”按钮，完成 C 语言源程序的调试和运行。C 语言编辑窗口如图 1-4 所示。

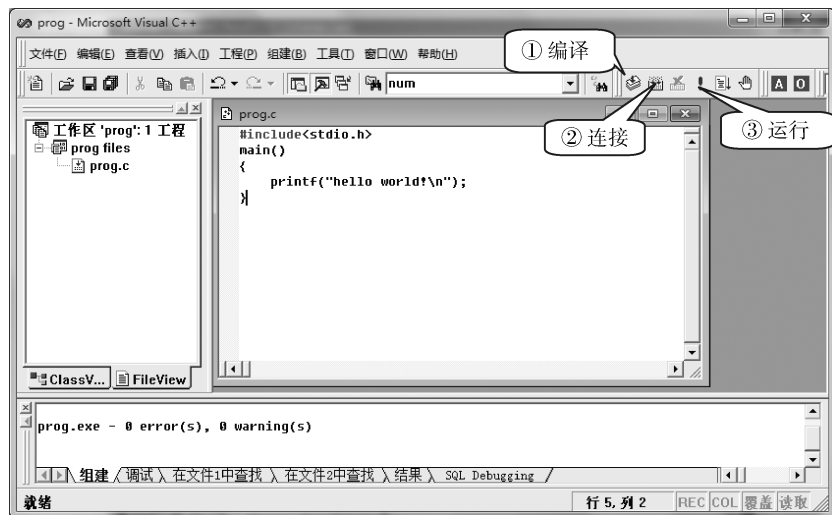


图 1-4 C 语言编辑窗口 (Visual C++ 6.0)

2) Visual C++ 2010 的编译环境

(1) 新建项目。Visual C++ 2010 程序，打开在文件菜单栏选择“新建”子菜单中的



“项目”命令，弹出“新建项目”对话框。在项目类型中单击“空项目”图标，选择该项目类型，在下方的“名称”文本框中输入项目名称，然后通过“位置”的“浏览”按钮选择存储路径。最后，单击“确定”按钮，完成项目的新建。在 Visual C++ 2010 的编译环境中创建 C 语言空项目的“新建项目”对话框如图 1-5 所示。



图 1-5 “新建项目”对话框 (Visual C++ 2010)

(2) 添加新项。在项目菜单栏选择“添加新项”命令，弹出“添加新项”对话框。在文件类型中单击“C++ 文件”图标，选择该文件类型，在下方的“名称”文本框中输入文件名，然后通过“位置”的“浏览”按钮选择存储路径。最后，单击“添加”按钮，完成文件的添加。“添加新项”对话框如图 1-6 所示。



图 1-6 “添加新项”对话框 (Visual C++ 2010)

(3) 调式和运行。在 C 语言编辑窗口输入代码后，先单击“生成”按钮，然后单击“生成解决方案”按钮，最后单击“运行”按钮，完成 C 语言源文件的调试和运行。C 语言编辑窗口如图 1-7 所示。



图 1-7 C 语言编辑窗口 (Visual C++ 2010)

在 Visual C++ 6.0 或在 Visual C++ 2010 编译环境中进行调试、运行时，可以使用快捷方式。其中，编译 (Compile) 的快捷方式为【Ctrl + F7】组合键，连接 (Build) 的快捷方式为【F7】快捷键，运行的快捷方式为【Ctrl + F5】组合键。

1.1.4 算法

1. 算法的概念

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。算法是程序的灵魂。

2. 算法的特性

一个算法具有以下特性：

1) 有穷性

一个算法应该包含有限的操作步骤。

2) 确定性

算法中的每一个步骤都应当是确定的。

3) 有零个或多个输入

一个算法既可以有多个输入，也可以没有输入。



4) 有一个或多个输出

一个算法的结果就是算法的输出。没有输出的算法是没有意义的。一个有效的算法可以没有输入，但是一定要有一个或者一个以上输出。

5) 有效性

算法中的每一个步骤都应当能有效执行，并得到确定的结果。

3. 算法的表示方法

算法的描述方法有很多，常见的有流程图和伪代码。其中，流程图比较形象直观。流程图的基本图形如图 1-8 所示。

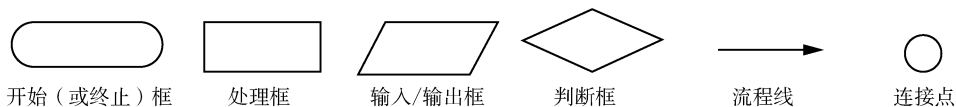


图 1-8 流程图的基本图形

4. 结构化程序设计

结构化程序设计由三种基本结构组成，分别是顺序结构、选择结构和循环结构。

1) 顺序结构

顺序结构是最简单的一种结构，按顺序执行相应的操作。若有两个任务，则先执行任务 A，再执行任务 B，其流程如图 1-9 所示。

2) 选择结构

选择结构是根据条件的成立与否来执行不同的操作。若有条件 P，当条件 P 成立时就执行任务 A，当条件 P 不成立时就执行任务 B，其流程如图 1-10 所示。

3) 循环结构

循环结构是判断条件，条件成立时重复执行某个操作，条件不成立时操作结束。若有条件 P，当条件 P 成立时就重复执行任务 A，当条件 P 不成立时就结束循环，其流程如图 1-11 所示。

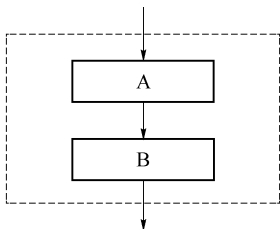


图 1-9 顺序结构流程

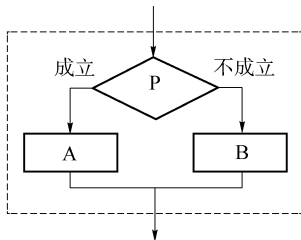


图 1-10 选择结构流程

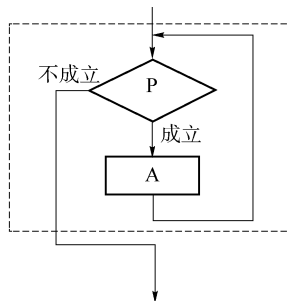


图 1-11 循环结构流程