



# C 语言程序设计

## ——程序思维与代码调试



周幸妮 著

 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>

# C 语言程序设计

——程序思维与代码调试

周幸妮 著

冯 磊 任智源 孙德春 参编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书站在程序设计的角度，从程序和算法、数据、程序语句、指针、结构体、函数和文件等基本的 C 语言要素讲起，全面介绍 C 语言程序设计的方方面面，引导学习者以“程序的思维”看问题，即如何从一个问题入手，算法应该如何设计、程序如何实现的角度去看程序设计问题，让初学者容易理解并掌握程序设计的基本思想与方法。通过对实际问题、解决方法或存在问题的讨论，引入新概念，深入浅出，让学习的过程变得有趣且容易。本书图文、表格并茂，便于直观理解。

本书不仅可作为普通高校本、专科学生及高职高专学生“C 语言程序设计”课程的教材或参考书，也可作为相关工程技术人员与自学者学习参考书。书后习题都有详细答案，程序题全部经过上机调试，并配有全套的微课视频。

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

C 语言程序设计：程序思维与代码调试 / 周幸妮著. — 北京：电子工业出版社，2019.10

ISBN 978-7-121-37098-4

I. ①C… II. ①周… 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 (2019) 第 144137 号

责任编辑：窦 昊

印 刷：三河市华成印务有限公司

装 订：三河市华成印务有限公司

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：787×1092 1/16 印张：29 字数：742.4 千字

版 次：2019 年 10 月第 1 版

印 次：2019 年 10 月第 1 次印刷

定 价：89.00 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 [zltts@phei.com.cn](mailto:zltts@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：(010) 88254466, [douhao@phei.com.cn](mailto:douhao@phei.com.cn)。

# 前言

## 缘起

2016 年底，电子工业出版社与中新金桥信息技术有限公司合作，计划做一批立体化网络课程建设项目，作者因有《C 语言程序设计新视角》（以下简称《新视角》）一书，有幸参与到项目之中。在 C 语言课程视频的录制过程当中，作者引入了许多新的思路与方法，因此有了对《新视角》进行完善再版的想法。

## 写作思路

《新视角》一书结合作者多年教学实践经验，针对学生在学习程序设计中的困惑，侧重引入问题、分析问题、讨论解决问题的方法，本书是基于《新视角》的改进版本。在 C 语言、数据结构等程序设计课程的讲授以及课后和学生的沟通中，笔者认识到在编程学习中大家遇到的普遍问题归纳起来有四难：概念晦涩理解难，规则众多记忆难，法无定法编程难，bug 深藏调试难。

著名数学家欧拉（Leonhard Euler）说过，如果不能把解决数学问题背后的思维过程传授给学生，那么数学教学就没有意义，这种观点同样适用于其他各学科或课程的教学。理解原理、掌握解决问题的思维方法是学习过程的要点。分析学生在学习程序设计中的问题，最主要的问题是编程概念建立困难，其次是程序调试技巧难以把握。按照问题轻重的顺序，《新视角》一书尝试从以下这些方面来解决问题——重思维，揭本质，强调试，轻语法。《新视角》经过这些年的使用，发现依然有不少需要改进的地方，特别是在问题引入和解析各种机制设置的原因方面，本书对此做了较大篇幅的增加。



## 1. 重思维

图灵奖获得者、现代计算机科学的先驱人物高德纳，在其代表作《计算机程序设计的艺术》中指出，编程是把问题的解法翻译成为计算机能“理解”的明确术语的过程，这是在人们开始试图使用计算机时最难以掌握的。C语言又是在高级编程语言中公认属于较难掌握的，有枯燥乏味、规则众多、艰深晦涩的一面。

计算机是一种自动化的工具，人们用这种工具来解决问题，会受到很多机器特性的限制，在一个与人们以往解决问题的运作规则不一样的系统中，已有的处理问题的经验很可能用不上。在程序设计语言中有很多概念是只学习过数理化知识的学生们从未接触过的，课堂上直接按传统教科书的内容讲述解释后，发现他们较难理解和接受。

Garr Reynolds 在《演说之禅》中这样写道：“好的故事可以终生受用，用于传道授业，用于分享，用于启迪，当然，也可以用于劝解。讲故事是吸引观众，满足他们对于逻辑、结构以及情感方面需求的一个重要方式。”本书中许多引例都是从有趣的故事开始，从实际问题引出与编程相关的话题，提出问题，引起读者思考，再从人直接解决问题到使用机器解决问题的不同角度加以讨论，分析相同与不同，最后提出相关的程序设计概念。为在叙述中有代入感，书中设计了布朗教授一家参与问题的讨论。布朗教授或以初学者的视角提出问题、探索解决方案，或以专家的身份讨论问题；布朗太太是编程小白，偶尔会有貌似可笑的外行言论；布朗教授的儿子小布朗是小学生，也常会提些看起来幼稚的问题。另外还有布朗教授的学生、同事及亲属参与“演出”。这样的设计思路，是遵循金出武雄（Takeo Kanade）教授在总结科研成功之道时提出的“像外行一样思考，像专家一样实践”方法论的一次实践。

“如果我们从不同的角度对同一个概念进行学习并研究与其相关的问题，就能建立更多且更深层次的信息链接。这些信息链接和与其相关的内容交织，共同构成了我们日常所说的‘理解’。大脑处理的所有信息都不是孤立的，而是具有逻辑关系的一组信息。当这些信息具有结构性，能够和已经学过的知识、和学生的生活实际建立密切的联系时，就容易让人从不同的角度来思考这一信息，从而对其有深入且透彻的理解。”（Salman Khan,《翻转课堂的可汗学院》）《新视角》通过解析人与计算机解决问题的同与不同，探索问题的解决方法，培养学生基于计算机特点进行逻辑思维的素养。根据计算机解决问题的特点，通过介绍读程列表法、算法设计法以及经典的算法描述法等方法，让学生先学读程再学编程，从大的方向上把握编程的一般方法，逐步建立程序的思维。

**Q** uestion

**法无定法编程难**

1

程序是怎么设计出来的?

**重思维**

**法无定法编程难 攻略**

**读程列表法** 学编程先把前人作品细观瞧 读程序祭起的法宝是列表

**算法设计法** 法无定法定编程依然法有道 先一般后特例异常须思考

**算法描述法** 结构化算法描述有经典指导 自顶向下逐步求精细化妙

## 2. 揭本质

“重视知识的本质，对于程序员来说这一点尤其重要。程序员的知识芜杂海量，而且总是在增长变化的。很多人感叹跟不上新技术。应对这个问题的办法只能是抓住不变量。大量的新技术其实只是一层皮，背后的支撑技术其实都是数十年不变的东西。底层知识永远都不过时，算法数据结构永远都不过时，基本的程序设计理论永远都不过时。”（刘未鹏，《暗时间》）

编程没有现成的公式可以套用，虽然学生通过大量实例的学习可以从中总结出一些规律，但对于程序设计语言为什么设这样或那样的处理规则，可能在短时间内不一定能琢磨明白。知其然还要知其所以然，只有清楚了各种规则设计的原理，才能彻底理解规则，进而熟悉和掌握规则，最后举一反三熟练应用。从原理的剖析入手，这样能提高学习效率，学会用计算机解决问题的方法。本书增加了许多从实际的问题引入概念的内容，让学生知道程序设计各重要概念的来龙去脉；增加了对重要机制的设置原因进行本质讨论分析，把有关联关系的概念进行横向或纵向分析链接，对重点的概念或相关联的概念提炼出其中的要素，以期增加学生对课程关键内容的把握。



## 3. 强调试

“无论一个程序的设计结构如何合理，也无论文档如何完备，如果不能产生正确的结果，则其一文不值。”（Chris H. Pappas 等所著的《C++程序调试》）人们制造的产品，一般都有一定的误差存在，比如设备或仪器。软件产品是一个例外，软件的最终交付形态是二进制的可执行码，执行码是不能容忍误差的，而人类的思维特性正好与之相反，是模糊和充满误差的，因此程序员很难一次就写出完全正确的代码。

张银奎在其《软件调试》一书中指出，“软件调试技术是解决复杂软件问题的最强大工具。如果把解决复杂软件问题看作一场战斗，那么软件调试技术便是一种可以直击要害而且锐不可当的武器。应该说学会调试器命令不难，但如何用调试器调试程序，找到 bug，却是一件很不容易的事情。”调试的技巧和经验需要很长时间的练习才能积累起来，对初学者而言是不容易掌握的，由此造成很多学生惧怕编程。

“我们可以在调试程序和侦破谋杀案之间找出相似点来。实际上，在几乎所有的谋杀悬念小



说中，谜案都是通过仔细分析线索，将表面上不重要的细节全联结起来而最终侦破的。”（Glenford J. Myers, 《软件测试的艺术》）“调试有点像捕鱼：相同的情绪、热情和刺激。长时间静静地工作后最终得到的是别人所不能体验的喜悦。”（Eugene Kotsuba）掌握了程序调试技术，读者在学习和开发程序时就能独立发现问题、解决问题，这样可以大大提高学习的兴趣和信心。

除了查找程序中的错误，调试其实对程序设计中不少概念的理解也是非常有帮助的，比如地址、存储单元、赋值、参数传递、作用域等，课堂上调试演示给予的直接的感性认识，比抽象的语言描述概念生动直观得多。“除了使用调试器调试程序、寻找代码中的问题，还可以认识其他软件、探索操作系统、观察硬件等。”（张银奎, 《软件调试》）不同的调试器，基本调试功能的工作方式和工作原理是基本一致的，“与 MS-DOS 中的第一个调试器 Debug.com 相比，雨后春笋般涌现的各种调试器，它们的大多数在原理上并没有多少进步，只是界面不同而已。”（Kaspersky, *Hacker Debugging Uncovered*）可以说，学会调试，对后续诸多的计算机类课程学习都是有益的，因此，教会学生调试的方法和技巧应该是编程类课程的一个非常重要的内容。

笔者在企业任程序员多年，参与过历时 4 年多、获国家科技进步奖的大型软件开发、开发成功后多用户单位安装调试、售后服务等系列工程工作，对程序测试与调试有一定的实践经验。在授课过程中一直坚持在课堂上穿插演示调试，学生只要有要求，随时可以将任何一个程序展开进行跟踪调试讲解。后来发现，即使在课上多次演示重要程序实例的调试，学生也掌握得并不好。细究其原因，调试是一个复杂过程，不同的数据组织、程序逻辑、现场查看处理等情形非简单几句描述就可以清楚明白，学生当时看懂了，但要回想复习时，就没有可以参考的纸面资料，所以在《新视角》一书中把各章很多重点实例的调试过程及技巧都“静态固化”到纸面上，让学习者学习调试时有切实可行的参照方法。调试内容千变万化，技巧也非常多，本书只讨论了最基本的调试方法与技巧，目前鲜有同类教科书有这样全面的处理方法，专门介绍调试的书籍也较少。

编程是一个不断修改才能完善的过程，其间往往要经过多次测试和调试。本书简单介绍了测试样例的设计原则和设计时机，让读者充分认识到程序测试的重要性，在学习编程之初就建立程序健壮性的概念。

**Q**uestio

**bug 深藏调试难**

3

Bug是如何Debug出来的?

**强 调试**

**bug 深藏调试难 之 攻略**

**程序测试** 程序是否合格需要测 设计测试样例有原则

**程序调试** 调试方法技巧费琢磨 用工具需要手眼配合  
多情形实例演示多多 勤思多练方能够掌握

#### 4. 轻语法

“轻”不是轻视语法，而是先介绍最基本的语法规则，让初学者在理解中记忆，在使用过程中逐步熟练，使学习者不至于一开始就被 C 语言繁复的规则所困惑。对于相关深入的语法或复杂的、不常用的规则，作者把握的原则是让学生知道知识归类、会根据线索自己查找说明手册或使用方法。针对计算机编程课程的特点，根据学生程度不同，由浅入深层层递进设置不同难度的练习。



#### 内容结构

本书探究从问题到解的计算机解决问题的方法论，以计算机解决问题的流程做主线，按照数据组织、数据处理、处理结果分别涉及的知识点层层展开，通过问题引入、类比解析概念，自顶向下逐步细化描述算法，数据分析实现存储、代码实现问题得解，到最后的测试调试验证结果。

数据的介绍从基本形式开始，随着问题要处理信息复杂程度的增加，通过讨论各种数据可能的组织方式，如数组、数据地址、复合数据、文件等，介绍数据在计算机中的组织结构和存储方法，另外还有数据的输入、输出方法。

算法是解决问题的步骤和方法，计算机算法需根据计算机解题特点来设计。算法由程序语句来实现，程序语句有相应语法规则及使用方法，程序有基本控制结构，程序开发有特定的步骤和方法。

随着问题复杂程度的增加，程序规模需要从单一模块（函数）变为多模块结构，本书通过讨论处理问题规模改变之后需要增加的处理机制，介绍函数的使用规则及相关问题。

在程序实现后，需要对代码进行测试，结果与预期不符则需要调试。本书介绍测试数据的设计原则、程序的运行环境，讨论调试方法。

本书习题丰富、解答详尽。从相对简单的热身题和基础题，到一般难度直至困难的作业题，让读者在循序渐进中保持学习的兴趣并不断进步。

#### 写作分工

本书共 12 章，周幸妮老师完成了第 1~9 章、第 12 章的写作及部分题目的整理；冯磊老师在原书基础上，对第 10 章、第 11 章进行重写，完成部分题目的设计；任智源老师完成所



有编程题目答案的编码调试工作，孙德春老师完成部分题目的收集整理等工作；全书由周幸妮老师统稿。

## 书稿说明

同许多高级程序语言相比，1972年诞生的C语言算是一种“古老”的语言，从ANSI C开始，经过不断修订，已经有一系列的标准。本书的C语言语法主要基于ANSI C，可能有些语法规则在最新的C99、C11标准中已经做了修改，但这些属于语法细节，并不影响书中的主要内容，根据“轻语法”的规则，书中未逐一按最新标准做修正。

书中例题分“例子”和“读程练习”两类。“例子”一般按照编程步骤，给出数据结构分析、算法描述、程序实现、跟踪调试等过程。“读程练习”一般只给出题目解释和参考程序，较少给出题目分析、程序实现过程解析，这些需要读者按照列表法等方法阅读理解程序。

为节省排版空间，书中代码格式可能部分缩紧，如花括号未单独占一行等。

本书所有例程在Visual C++ 6.0集成环境中调试通过。Visual C++ 6.0虽然是比较老的运行调试环境，但依然是目前国家计算机等级考试要求的C语言运行环境，且安装容量小，系统兼容性较好；再者，调试的基本思想具有普适性，并不局限于一种程序设计语言和某种运行环境。

## 致谢

除了《C语言程序设计新视角》一书中致谢提及的屈宇澄、屠仁龙、孙蒙等同学，还要感谢在《新视角》一书使用中提出意见和建议的同事和同学们，他们促使作者反思《新视角》中的缺陷和不足，从心理学、认知规律等各方面做更多的学习和思考，加强问题的引入，改变叙述方式等，这些改进在后续出版的《数据结构与算法分析新视角》中获得了较好的效果。感谢为本书提供帮助的丁煜、郭丽萍、宋昀翀等同学。感谢中新金桥信息技术有限公司支持的课程建设项目，使作者从中学习了很多微课制作的知识和技巧，与网络时代共同进步。

《新视角》历久积累再次重写，宛若生命轮回赤子新生，借2019年初春感言是以为记：春日花繁，生命绽放，雯华若锦，可喜可贺。

周幸妮

xnzhou@xidian.edu.cn

2019年仲夏于长安



# 目 录

第 1 章 程序概论	1	第 3 章 基本数据	54
1.1 流程的概念	1	3.1 常量与变量	54
1.1.1 关于流程	1	3.1.1 常量	54
1.1.2 流程的表达方式	3	3.1.2 变量	56
1.1.3 流程的基本逻辑结构	4	3.2 数据类型	62
1.2 程序的概念	8	3.2.1 计算机中的信息表示	62
1.2.1 自动化流程	9	3.2.2 计算机中的信息处理问题 讨论	63
1.2.2 程序的概念	9	3.2.3 C 语言的基本数据类型	66
1.2.3 程序的执行特点	12	3.3 整数存储规则	67
1.2.4 计算机工作流程	12	3.3.1 有符号整数	68
1.3 程序的构成	13	3.3.2 无符号整数	68
1.3.1 计算机解题流程之数据	14	3.3.3 字符类型数据	69
1.3.2 计算机解题流程之处理	15	3.4 实数存储规则	70
1.3.3 计算机解题流程之结果	15	3.5 运算符与表达式	74
1.4 程序的开发过程	17	3.6 数值处理	76
1.4.1 问题引例	17	3.6.1 算术运算符和算术表达式	76
1.4.2 程序开发基本步骤	18	3.6.2 数据运算中的出界问题	80
1.4.3 计算机解题实例	19	3.7 逻辑判断处理	81
1.4.4 程序开发流程	24	3.7.1 关系运算	81
1.5 C 语言程序简介	25	3.7.2 逻辑运算	82
1.5.1 C 程序样例	25	3.8 数据类型转换	86
1.5.2 C 程序框架结构	28	3.8.1 强制类型转换	88
1.5.3 代码格式要求	29	3.8.2 自动类型转换	89
1.6 本章小结	30	3.9 其他运算	90
习题	31	3.9.1 条件表达式	90
第 2 章 算法	32	3.9.2 sizeof 运算符	91
2.1 算法的概念	32	3.9.3 赋值运算符与表达式	92
2.2 算法的表示	35	3.9.4 复合赋值运算符	92
2.3 算法的可行性	39	3.9.5 逗号运算符和逗号表达式	92
2.4 算法的通用性	42	3.10 本章小结	93
2.5 算法的全面性	44	习题	94
2.6 算法设计过程与算法特性	49	第 4 章 输入/输出	95
2.6 本章小结	52	4.1 输入/输出的概念	95
习题	52		

4.1.1	标准输入/输出	95	5.6.3	do-while 语句实例	150
4.1.2	C 标准库函数	96	5.7	当型循环的另一种形式	151
4.1.3	头文件	96	5.7.1	for 语句语法规则	151
4.2	数据的输出	97	5.7.2	for 语句实例	152
4.2.1	字符输出函数	97	5.8	无限循环	154
4.2.2	字符串输出函数	98	5.8.1	实际问题中的无限制循环	154
4.2.3	格式输出函数	99	5.8.2	无限循环的 while 语句表达	154
4.3	数据的输入	102	5.8.3	无限循环的 for 语句表达	155
4.3.1	字符输入函数	103	5.9	中断循环	157
4.3.2	字符串输入函数	104	5.9.1	实际问题中的循环中断	157
4.3.3	格式输入函数	105	5.9.2	跳出循环的 break 语句	158
4.4	数据输入的常见问题	109	5.9.3	在循环内跳转的 continue 语句	161
4.5	本章小结	112	5.10	自由跳转机制	164
	习题	113	5.10.1	自由跳转的概念	164
<b>第 5 章</b>	<b>程序语句</b>	115	5.10.2	无条件转移语句规则	164
5.1	顺序结构	115	5.10.3	无条件转移语句实例	164
5.2	双分支选择结构	117	5.10.4	goto 语句的特点	166
5.2.1	双分支选择结构的语法规则	117	5.11	本章小结	167
5.2.2	复合语句的作用	118		习题	168
5.2.3	if 语句实例	119	<b>第 6 章</b>	<b>数组</b>	172
5.2.4	嵌套的 if-else 语句	121	6.1	数组的概念	172
5.3	多分支选择结构	124	6.1.1	一组同类型数据的处理问题	172
5.3.1	多分支问题的引入	124	6.1.2	一组同类型数据所需要的表达方式	175
5.3.2	多分支结构语法规则	125	6.2	数组的存储	176
5.3.3	多分支结构实例	128	6.2.1	数组的定义	176
5.3.4	各种分支结构语句的比较	134	6.2.2	数组的初始化	178
5.4	循环问题的引入	134	6.2.3	数组的空间分配	179
5.4.1	循环中的要素分析	134	6.2.4	数组的空间查看	180
5.4.2	循环三要素	136	6.3	一维数组的操作	183
5.4.3	循环语句	137	6.4	二维数组的操作	196
5.5	当型循环结构	137	6.5	字符数组的操作	205
5.5.1	当型循环语法规则	137	6.6	本章小结	214
5.5.2	循环要素必要性验证	138		习题	215
5.5.3	当型循环实例	142	<b>第 7 章</b>	<b>指针</b>	217
5.5.4	循环控制方式	146	7.1	指针的概念	217
5.6	直到型循环结构	146			
5.6.1	直到型循环语法规则	146			
5.6.2	do-while 的适用场合	150			

7.1.1	名称引用和地址引用	217	8.6.2	typedef 声明形式及使用	275
7.1.2	存储空间的管理	219	8.7	本章小结	275
7.1.3	指针的概念	223	习题		276
7.2	指针的运算	224	<b>第 9 章 函数</b>		278
7.2.1	指针运算符	224	9.1	函数的概念	278
7.2.2	指针运算种类	224	9.1.1	问题的提出	278
7.2.3	指针运算基本规则	224	9.1.2	模块的概念	279
7.2.4	指针偏移的意义	227	9.2	函数形式设计	281
7.2.5	空指针的概念	230	9.2.1	模块间信息交流方法	281
7.3	指针与数组	230	9.2.2	函数形式设计	282
7.3.1	指针与一维数组	230	9.3	函数间信息交流机制设计	285
7.3.2	指针与二维数组	234	9.3.1	函数间信息交流特点分析	285
7.4	指针与多组字符串问题	238	9.3.2	函数间信息交流之处理数据的提交与接收	287
7.4.1	一维指针数组与指向指针的指针	240	9.3.3	函数结果的获取方式	288
7.5	本章小结	241	9.4	函数总体设计	289
习题		242	9.4.1	函数设计要素	289
<b>第 8 章 复合类型数据</b>		244	9.4.2	函数间信息传递归结	289
8.1	结构体的概念	244	9.4.3	函数的调用	290
8.1.1	问题引入	244	9.5	函数设计实例	292
8.1.2	综合数据表的存储方案	245	9.5.1	传值调用	292
8.2	结构体的存储	246	9.5.2	传址调用	298
8.2.1	结构体类型定义	246	9.5.3	函数综合实例	306
8.2.2	结构体变量定义	248	9.5.4	main 函数的参数	316
8.2.3	结构体初始化	248	9.6	作用域	319
8.2.4	结构体变量空间分配	249	9.6.1	问题引入	319
8.2.5	结构体成员引用	253	9.6.2	模块的屏蔽机制	321
8.3	结构体应用实例	254	9.6.3	内存分区与存储分类	322
8.4	共用体	264	9.6.4	屏蔽机制 1——变量的有效期和作用范围	323
8.4.1	问题引入	264	9.6.5	屏蔽机制 2——函数的有效范围	330
8.4.2	共用体的空间存储描述	264	9.6.6	屏蔽机制 3——共享数据的使用限制	332
8.5	枚举	269	9.7	递归	333
8.5.1	问题引入	269	9.7.1	引例	333
8.5.2	枚举的概念及定义形式	270	9.7.2	递归概念	336
8.5.3	枚举实例	271	9.7.3	递归实例	337
8.5.4	枚举的使用规则	272			
8.6	声明新的类型名	273			
8.6.1	问题引入	273			

9.8 本章小结	339	12.1.7 运行程序	386
习题	340	12.2 程序测试	387
<b>第 10 章 编译预处理——编译前的</b>		12.2.1 引子	387
<b>工作</b>	343	12.2.2 程序测试方法与实例	388
10.1 问题的引入	343	12.3 程序调试概念	392
10.2 宏定义	344	12.3.1 bug 与 debug	392
10.2.1 简单的宏定义	344	12.3.2 bug 无处不在	393
10.2.2 带参数的宏定义	346	12.3.3 软件调试的困难	393
10.2.3 宏定义的副作用	348	12.4 软件调试的方法论	394
10.3 文件包含	348	12.4.1 引例	394
10.4 条件编译	350	12.4.2 软件调试的基本过程	395
10.5 本章小结	353	12.4.3 程序错误的查找方法讨论	395
习题	354	12.4.4 跟踪方法方案探索	397
<b>第 11 章 文件——外存数据的操纵</b>	356	12.5 程序调试工具	399
11.1 问题的引入	356	12.5.1 IDE 中调试器的功能	399
11.2 文件的概念	357	12.5.2 调试命令	401
11.3 文件的操作流程	358	12.6 调试实例	405
11.4 内存和外存的数据交流	359	12.6.1 基本调试步骤示例	405
11.5 程序对文件的操作	361	12.6.2 调试查找程序错误示例	407
11.5.1 打开文件	361	12.6.3 调用栈的使用示例	416
11.5.2 文件的读写	362	12.6.4 数据断点使用示例	418
11.5.3 关闭文件	366	12.7 本章小结	420
11.5.4 随机读取文件内容	367	习题	421
11.6 关于文件读写函数的讨论	368	附录 A 运算符的优先级和结合性	425
11.7 程序调试与输入输出重定向	372	附录 B ASCII 码表	426
11.8 本章小结	374	附录 C C 语言常用库函数	427
习题	375	附录 D 常用转义字符表	432
<b>第 12 章 程序的运行</b>	377	附录 E 位运算简介	433
12.1 程序运行环境	377	附录 F 在工程中加入多个文件	435
12.1.1 集成环境主界面	379	附录 G 编程范式	441
12.1.2 建立项目	380	附录 H 空类型 void 问题	449
12.1.3 新建源文件	382	参考文献	450
12.1.4 编辑源文件	382		
12.1.5 编译源文件	384		
12.1.6 链接程序	385		

# 第 1 章 程序 概 论

## 【主要内容】

- 流程的概念；
- 程序的概念；
- 程序设计方法；
- 简单的 C 语言程序介绍。

## 【学习目标】

- 掌握程序的概念；
- 理解程序设计的基本步骤；
- 了解 C 语言程序的基本框架结构。

## 1.1 流程的概念

人们使用计算机来帮助自己高效地工作。计算机是如何工作的呢？回答这个问题，我们要先从人处理问题的过程入手，考查解决问题的思路、方法和手段。

### 1.1.1 关于流程

我们先来看一些实际生活中的流程。

某高校的开学典礼有若干环节，如图 1.1 所示，其中时序就是时间的先后顺序，这里用带箭头的线表示。把各个环节按完成时间的先后顺序排列起来，这样形成的一个过程流，就叫做流程。

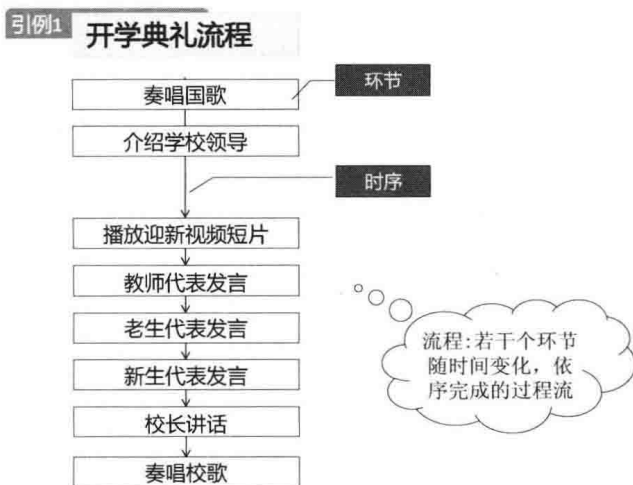


图 1.1 开学典礼流程



很多同学出行都坐过火车，有购买火车票的经验。图 1.2 中列出了通过窗口购票的具体步骤。可以看出，购票流程是预定一个任务，通过具体执行步骤，达到设定目标的一个过程描述。

馒头是我国北方常见的主食之一，馒头的制作是一个比较复杂的工艺过程，图 1.3 列出了其主要制作流程。人们将各种原材料通过一定的设备、按照一定的顺序进行加工，最后制成成品的方法与过程就是制作流程。

### 引例2 售票处购火车票流程

- 第1步：旅客提供出行日期、目的地地点等信息  
 第2步：售票员找到相应日期里可以供选择的车次  
 第3步：旅客确定车次、购票数量  
 第4步：旅客付款、拿票

图 1.2 窗口购票流程

### 引例3 馒头制作流程

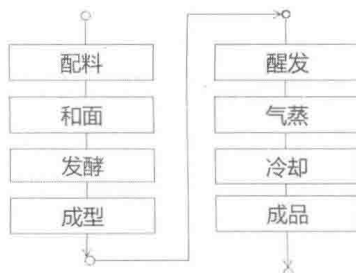


图 1.3 馒头制作流程

现在出远门旅行，越来越多的人选择了乘坐飞机，图 1.4 所示的登机流程，对首次坐飞机的乘客有清晰的指引作用。

### 引例4 登机流程

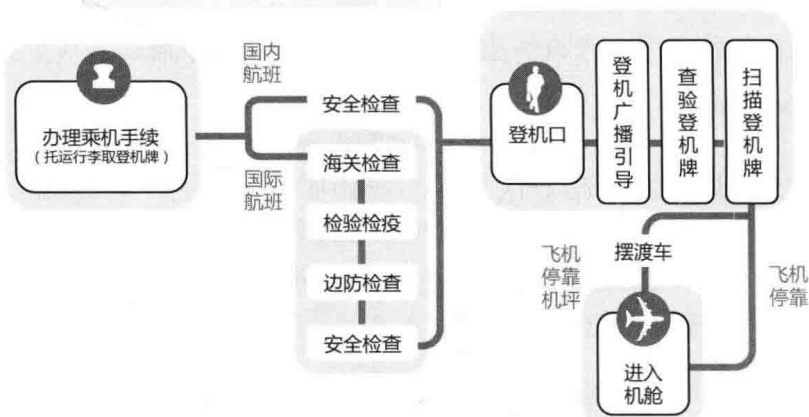


图 1.4 机场登机流程

通过前面的例子，我们可以看出，不论是工作流程还是工艺流程，其目的都是实现特定的结果或产品，它反映完成一项任务、一件事情或者生产、制造某种产品的全过程。不管什么流程，它们的共同特点是由一系列的环节和时序组成，如图 1.5 所示。

### 流程

流程反映完成一件事情、一项任务或生产制造某种产品的全过程。流程由一系列的环节和时序组成。

**环节：**完成某个具体目标、组成某项生产或某个活动过程若干阶段或小的过程。  
**时序：**流程中各个环节在时间上的先后顺序。

图 1.5 流程的概念

### 1.1.2 流程的表达方式

从前面的例子我们可以看到，流程的表达方式是多种多样的，开学典礼的过程描述，是以流程图的形式；购火车票的流程描述，是用文字表述的；乘机流程采用的是图的解说形式；另外流程还可以用表格、模型、视频等形式来表示。程序设计中经常采用流程图和伪代码等方式描述流程，如图 1.6 所示。

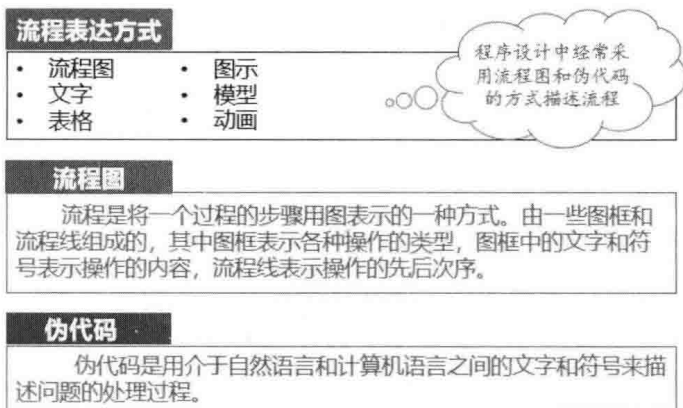


图 1.6 程序设计中常用流程表示方法

流程图是将一个过程步骤用图表示的一种方式。伪代码是用文字和符号来描述处理过程，它不用图形符号，书写方便、格式紧凑、易懂，便于向程序过渡。

本书主要采用伪代码来描述程序处理流程，人们处理问题的方法步骤也称为“算法”。

#### 1. 流程图

美国国家标准化协会（American National Standard Institute, ANSI）规定了一些常用的流程图符号，为世界各国程序工作者普遍采用。最常用的流程图符号如图 1.7 所示。其中处理框、判断框、输入/输出框都是用来表示各种情形的环节，流程线是一条带箭头的线段，用来表示时序。用图形表示流程，直观形象，易于理解。

符号	名称	意义	
○	起止框	表示流程的开始或结束	} 环节
□	处理框	表示一般的处理功能	
◇	决策/判断框	对一个给定的条件进行判断，结果为是或否。要在出口处标明判断结果	
▱	输入输出框	数据的输入输出	
→	流程线	连接处理或判断框，指明流程的路径和方向	} 时序
○	连接点	将画在不同地方的流程线连接起来	

图 1.7 常用流程图符号

#### 2. 伪代码

伪代码（Pseudocode）：用代码的格式表示程序执行过程和算法，伪代码是不依赖于具体

程序语言的，它只是用程序语言的结构形式来表示程序执行，因此不能在编译器上编译。其目的是用易于理解和表述的方式展示程序的执行过程。

### 【知识 ABC】 流程图与伪代码

从 20 世纪 40 年代末到 70 年代中期，程序流程图一直是过程设计的主要工具。它的主要优点有采用简单规范的符号，画法简单；结构清晰，逻辑性强；便于描述，容易理解，是对控制流程的描绘很直观，便于初学者掌握。由于程序流程图历史悠久，为人所熟悉，尽管它有种种缺点，许多人建议停止使用它，但至今仍在广泛使用着。程序流程图是人们对解决问题的方法、思路或算法的一种描述。不过总的趋势是越来越多的人不再使用程序流程图了。程序流程图的主要缺点一是所占篇幅较大，由于允许使用流程线，过于灵活，不受约束，使用者可使流程任意转向，从而造成程序阅读和修改上的困难。二是不利于结构化程序的设计，程序流程图本质上不是逐步求精的好工具，它会使程序员过早地考虑程序的控制流程，而不去考虑程序的全局结构。

人们在用不同的编程语言实现同一个算法时意识到，它们的实现（注意，是实现，不是功能）往往是不同的，尤其是对于那些熟练于某种编程语言的程序员来说，要理解一个用其他编程语言编写的程序的功能可能很困难，因为程序语言的形式限制了程序员对程序关键部分的理解。伪代码就这样应运而生了。

当考虑算法功能（而不是其语言实现）时，常常应用伪代码。计算机科学在教学中通常使用伪代码，以使所有程序员都能理解。

伪代码以编程语言的书写形式指明算法的职能。相比于程序语言（如 Java、C++、C、Delphi 等）它更类似自然语言。可以将整个算法运行过程的结构用接近自然语言的形式描述出来。这里用户可以使用任何一种熟悉的文字，如中文或英文等，关键是把程序的意思表达出来。使用伪代码，可以帮助我们更好地表述算法，不用拘泥于具体的实现。

#### 1.1.3 流程的基本逻辑结构

处理问题的流程描述有哪些逻辑特性呢？下面我们就来讨论这个问题。

先来看一个实际生活中的流程问题——洗衣机的设置。

##### 1. 顺序结构

布朗先生是大学的计算机教授，平时工作比较忙。布朗太太近期要出门一段时间，临走前要教会教授使用家中的洗衣机，洗衣机的功能很多，她怕说得复杂了，布朗先生学不会，所以只教了最基本的设置，如图 1.8 所示。

洗衣服的各步骤按序逐条执行，顺序的先后由步骤的逻辑关系决定，比如应该先浸泡，再洗涤。布朗先生认真做了笔记，按洗衣步骤的先后顺序画了一个洗衣步骤流程图，如图 1.9 所示。在这个洗衣功能中，预设指令是按顺序排列的，因此这样的操作指令的组合结构被称为“顺序结构”。

##### 2. 判断结构

布朗先生第一次用洗衣机，放了大量衣物，按太太教的步骤设置好程序，启动后发现洗