



“十二五”普通高等教育本科国家级规划教材  
国家级精品资源共享课和国家精品在线开放课程主讲教材

“高等学校本科计算机类专业应用型人才培养研究”项目规划教材

# C 语言程序设计 (第4版)

**Programming in C (Forth Edition)**

苏小红 赵玲玲 孙志岗 王宇颖 等编著  
蒋宗礼 主审

高等教育出版社



“十二五”普通高等教育本科国家级规划教材  
国家级精品资源共享课和国家精品在线开放课程主讲教材

“高等学校本科计算机类专业应用型人才培养研究”项目规划教材

# C 语言程序设计 (第4版)

Programming in C (Forth Edition)

苏小红 赵玲玲 孙志岗 王宇颖 等编著  
蒋宗礼 主审

高等教育出版社·北京

## 内容简介

本书是一本兼具趣味性和实用性的 C 语言程序设计教材。全书由 14 章组成,内容包括:为什么要学习编程,基本数据类型,基本算术运算,键盘输入和屏幕输出,选择控制结构,循环控制结构,函数与模块化程序设计,数组和算法基础,指针,字符串,指针和数组,结构体和数据结构基础,文件操作以及简单的游戏设计。

本书以应用为背景,面向编程实践和问题求解能力的训练,从实际问题出发,在案例的不断深化中逐步引出知识点,形成“程序设计方法由自底向上到自顶向下”和“数据结构由简单到复杂”的两条逻辑清晰的主线。案例内容紧密结合实践,举一反三,融会贯通,尤其侧重错误案例的分析和讲解,在任务驱动下,由浅入深、启发引导读者循序渐进地编写规模逐渐加大的程序,让读者在不知不觉中逐步加深对 C 语言程序设计方法的了解和掌握。本教材配有多媒体课件、例题和习题源代码以及基于 C/S 结构的编程题考试自动评分系统等教学资源,免费向任课教师提供。

本书适合作为高等学校各专业的 C 语言程序设计课程教材,也可作为从事计算机相关工作的科技人员、计算机爱好者及各类自学人员参考。

## 图书在版编目(CIP)数据

C 语言程序设计 / 苏小红等编著;蒋宗礼主审. --  
4 版. -- 北京:高等教育出版社,2019.8  
ISBN 978-7-04-051998-3

I. ①C… II. ①苏… ②蒋… III. ①C 语言-程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 097617 号

策划编辑 刘茜      责任编辑 刘茜      封面设计 张志      版式设计 张杰  
插图绘制 于博      责任校对 刘莉      责任印制 尤静

---

出版发行	高等教育出版社	网 址	<a href="http://www.hep.edu.cn">http://www.hep.edu.cn</a>
社 址	北京市西城区德外大街 4 号		<a href="http://www.hep.com.cn">http://www.hep.com.cn</a>
邮政编码	100120	网上订购	<a href="http://www.hepmall.com.cn">http://www.hepmall.com.cn</a>
印 刷	北京明月印务有限责任公司		<a href="http://www.hepmall.com">http://www.hepmall.com</a>
开 本	787mm×1092mm 1/16		<a href="http://www.hepmall.cn">http://www.hepmall.cn</a>
印 张	29.5	版 次	2011 年 4 月第 1 版
字 数	670 千字		2019 年 8 月第 4 版
购书热线	010-58581118	印 次	2019 年 8 月第 1 次印刷
咨询电话	400-810-0598	定 价	49.00 元

---

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究

物料号 51998-00

# C 语言程序设计 (第4版)

苏小红 赵玲玲 孙志岗

王宇颖 等编著

- 1 计算机访问<http://abook.hep.com.cn/1860423>, 或手机扫描二维码、下载并安装 Abook 应用。
- 2 注册并登录, 进入“我的课程”。
- 3 输入封底数字课程账号(20位密码, 刮开涂层可见), 或通过 Abook 应用扫描封底数字课程账号二维码, 完成课程绑定。
- 4 单击“进入课程”按钮, 开始本数字课程的学习。



课程绑定后一年为数字课程使用有效期。受硬件限制, 部分内容无法在手机端显示, 请按提示通过计算机访问学习。

如有使用问题, 请发邮件至[abook@hep.com.cn](mailto:abook@hep.com.cn)。



扫描二维码  
下载 Abook 应用



程序设计实践教学软件介绍

<http://abook.hep.com.cn/1860423>

## 出版说明

信息化社会需要大量的计算机类专业人才。据统计,目前我国计算机类专业布点总数已逾 2 800 个,这些专业点为国家的现代化建设培养了大批计算机类专业人才,其中绝大多数是应用型人才。如何按照社会需求,确定合理的人才培养目标,并在其“制导”下培养特色突出的应用型人才,是提高教育质量和水平的重要任务。

为了更好地引导高校计算机类各专业点构建有特色的培养方案,例如,能够体现行业特色、区域需求,同时建设体现这些特色的学科基础课和专业课,促进本科计算机类专业应用型人才培养,出版一批体现应用型人才培养特色的新形态教材,教育部高等学校计算机类专业教学指导委员会、全国高等学校计算机教育研究会与高等教育出版社联合组建了“高等学校本科计算机类专业应用型人才培养研究”课题组,基于《计算机类专业教学质量国家标准》,围绕软件工程、网络工程、物联网工程等专业应用型人才培养的研究展开相关工作。

在研究的基础上,课题组汇聚 80 多所高校的教学经验,协同创新,开展了核心课程教学资源建设以及教材建设,这套教材作为课题研究的重要成果之一,具有以下几个显著特点。

- 以课题研制的《高等学校本科计算机类专业应用型人才培养指导意见》为指导,委托有丰富的教学实践经验的教师编写,内容覆盖了不同专业的学科基础课、专业核心课及专业方向课。
- 教材内容基于理论适用,突出理论与实践相结合,强调“做中学”,引入丰富的实验案例,摒弃大而全、重理论轻实践的做法,结构新颖、努力突出专业特色。
- 采用纸质教材与数字资源相结合的形式,将教学内容与课程建设充分展示出来,使教师和学生借助网络实现全方位的个性化教学。

相信这套教材的出版能够起到推动各高校计算机类专业建设、提高教学水平和人才培养质量的作用。希望广大教师在教学过程中对教材提出宝贵的意见和建议,使其在使用过程中不断完善。

教育部高等学校计算机类专业教学指导委员会  
全国高等学校计算机教育研究会  
高等教育出版社

# 序

程序设计是一门非常重要的课程,其重要性不仅仅体现在一般意义上的程序编制,更体现在引导读者实现问题求解思维方式的转换——培养计算思维能力。也正是由于需要引导初学者实现思维方式的转换,才使得这门看似简单的课程具有很高的难度,突破了这个难点,一切将变得比较自然。这本教材以C语言为背景,从初学者的需求出发,在面向工程应用型计算机专业人才的培养方面进行了有益的探索,体现了“学生易学,教师易用,变应试为应用”的编写理念,形成了如下一些特点。

1. 以实际问题的求解过程为引导,讲授程序设计的基本方法,以结构化与模块化程序设计为核心,沿着数据结构从简单到复杂这条线逐步展开,侧重对程序设计方法、程序调试方法的介绍,并将软件工程相关的思想和方法渗透其中,提高读者程序编制的规范性。

2. 重点放在解决“程序设计”的核心问题上,以讲授程序设计为主,将C语言的有关语法有机地结合到程序设计中,避免了生硬枯燥的语法叙述,真正体现了“程序设计”,在“把C语言从应试课程转变为实践工具”上做出了可贵的探索。

3. 明显地体现出作者多年来在该门课程上的教学积累,在写作上努力追求面向初学者进行“讲授”的风格,行文流畅,语言带有一定的人文气息,努力贴近读者,深入浅出,通俗易懂,逻辑性强,形成该书独特的风格。

4. 将作者丰富的程序设计经验融入教材编写,按照初学者的需求,适时引导进行程序错误分析、测试与调试,将一些容易被忽略的而且对高水平C语言程序设计很重要的“点”逐一展现给读者,进一步落实“程序设计”教学的需求。

5. 选择了一些趣味性强、有吸引力的例子和话题以提高读者的学习兴趣,选择一些实用性强的例子和话题,以努力提高读者的工程实践能力。精选的“不断提升”的引导性例题、习题和实验题,以及贯穿全书的综合实例,起到了开拓读者思路、引导读者探究问题求解方法、激发读者程序设计兴趣的作用。

6. 按照教学的需要,本书还配套建设了丰富的教学资源,如《C语言程序设计学习指导》、程序源代码、多媒体课件、编程题考试自动评分系统、学习自测软件以及课程教学网站等,构成了C语言程序设计课程教学的完整解决方案。

希望该书能够得到众多读者的喜爱!



教育部高等学校计算机类专业教学指导委员会副主任委员  
全国高等学校计算机教育研究会理事长

# 前 言

我们所使用的工具深刻地影响着我们的思维方式和思维习惯,进而也将深刻地影响着我们的思维能力。

——Edsger W.Dijkstra

学习程序设计本身是一件既充满挑战、更充满乐趣的事情。然而,它常常会给人以枯燥乏味的感觉,是因为没有人帮助发掘出其中的趣味。本书力图用最简明的语言、最典型的实例以及最通俗的类比和解释将这种趣味性挖掘出来,带给读者全新的学习体验,和读者一起欣赏 C 语言之美,领悟 C 语言之妙,体会学习 C 语言之无穷乐趣。

本书以应用为背景,面向编程实践和问题求解能力训练,从实际问题出发,在一个实际案例的不断深化中逐步引出相关知识点,借助任务驱动的实例将相关知识点像珠链一样串联起来,形成“程序设计方法由自底向上到自顶向下”和“数据结构由简单到复杂”的两条逻辑清晰的主线。案例内容紧密结合实践,举一反三,融会贯通。在任务驱动下,由浅入深、启发引导读者循序渐进地编写规模逐渐加大的程序,让读者在不知不觉中逐步加深对 C 语言程序设计方法的了解和掌握。

在内容的指导思想,本书以 C 语言为工具,介绍计算思维方法和程序设计的基本方法,不拘泥于 C 语言的基本语法知识,面向实际应用,把计算思维方法和程序设计中最基本、最新、最有价值的思想和方法渗透到 C 语言的介绍中。目的是使读者在学习了 C 语言以后,无论使用什么语言编程,都具有灵活应用这些思想和方法的能力。

全书共分 14 章,内容包括:为什么要学习编程,基本数据类型,基本算术运算,键盘输入和屏幕输出,选择控制结构,循环控制结构,函数与模块化程序设计,数组和算法基础,指针,字符串,指针和数组,结构体和数据结构基础,文件操作以及简单的游戏设计。

在内容编排上,全书内容注重教材的易用性。每章开头都有内容导读,指导读者阅读,每章结尾给出本章知识点小结和常见错误小结,帮助读者整理思路。本书既适合于程序设计的初学者,也适合于想更深入了解 C 语言的人。书中设计了很多思考题,并在每章的扩充内容中增加了一些有一定深度和开放性的内容,供希望深入学习程序设计的读者选学和参考,力求做到内容有宽度、有深度、有高度。

在内容写作上,本书力图避免以往教材编写中常常出现的通病和问题,如“实例不实,为解释语法而设计”、“语法堆砌,只见树木不见森林”、“忽视错误程序的分析 and 讲解”等。因此编写的主要特色是注重错误程序的讲解和分析以及与软件工程内容的联系。在分析常见错误案例

的过程中,讲解程序设计的基本方法、程序测试方法以及程序调试和排错方法,帮助读者了解错误发生的原因、实质、排错方法及解决对策。

全书程序采用统一的代码规范编写,并且在编码中注重程序的健壮性。全书例题、习题和实验题的内容选取兼具趣味性和实用性,习题以巩固基本知识点和强化程序设计方法为目的,难度分成多个阶梯,包括:改写例题的编程题,模仿例题的编程题,趣味游戏类编程题。题型包括:侧重程序阅读理解能力训练的写出程序运行结果题和程序填空题,侧重程序调试和排错能力训练的分析改错题,侧重编程实践能力训练的任务递进式编程题等。本书第 14 章给出了迷宫和 Flappy bird 两个游戏实例的程序设计,配套的学习指导给出了一个综合应用程序“学生成绩管理系统”的程序设计,以及 2048、贪吃蛇和扫雷游戏的程序设计。

本书是爱课程网中国大学 MOOC 平台上开设的国家精品在线开放课程“C 语言程序设计精髓”和国家精品资源共享课程“C 语言程序设计”的主讲教材,与教材配套的教学资源包括:

- (1) 面向读者的教材网站:<http://book.sunner.cn>,可下载本书例题的源代码和最新勘误表。
- (2) 爱课程网“中国大学 MOOC”频道:搜索“C 语言程序设计精髓”课程,可观看课程的微视频,浏览课件和算法演示动画,进行编程题在线测试,与老师和同学在线讨论。
- (3) 爱课程网“资源共享课”频道:搜索“C 语言程序设计”课程,可观看课程的全程教学录像,浏览课件和算法演示动画,进行课程评价或提问。
- (4) 基于 C/S 结构的 C 语言编程题考试自动评分系统(签署软件使用协议后可免费获取)。
- (5) 基于 B/S 结构的 C 语言编程题考试自动评分系统。
- (6) 基于 B/S 结构的 C 语言试卷与题库管理系统。
- (7) 面向学生自主学习的 C 语言在线作业和能力测试系统(使用教材封四的刮刮卡可获得有效期一年的注册用户名,支持读者和学生在线完成本书章后习题并获得系统自动评测结果。
- (8) Code::Blocks+gcc+gdb 安装程序可在 Code::Blocks 官网下载。
- (9) 扫描本书的二维码,可浏览教学课件和算法演示动画,以及观看部分教学视频。

上述各种教学资源之间的关系及介绍如下图所示。有需要者可直接与作者本人联系([sxh@hit.edu.cn](mailto:sxh@hit.edu.cn))。

与本书配套出版的《C 语言程序设计学习指导(第 4 版)》,主要包括习题解答、实验指导和编程题目与解答三部分内容。实验指导中介绍了在 Code::Blocks+gcc+gdb 环境、Visual Studio 以及 Clion 等集成开发环境下的标准 C 程序调试方法。



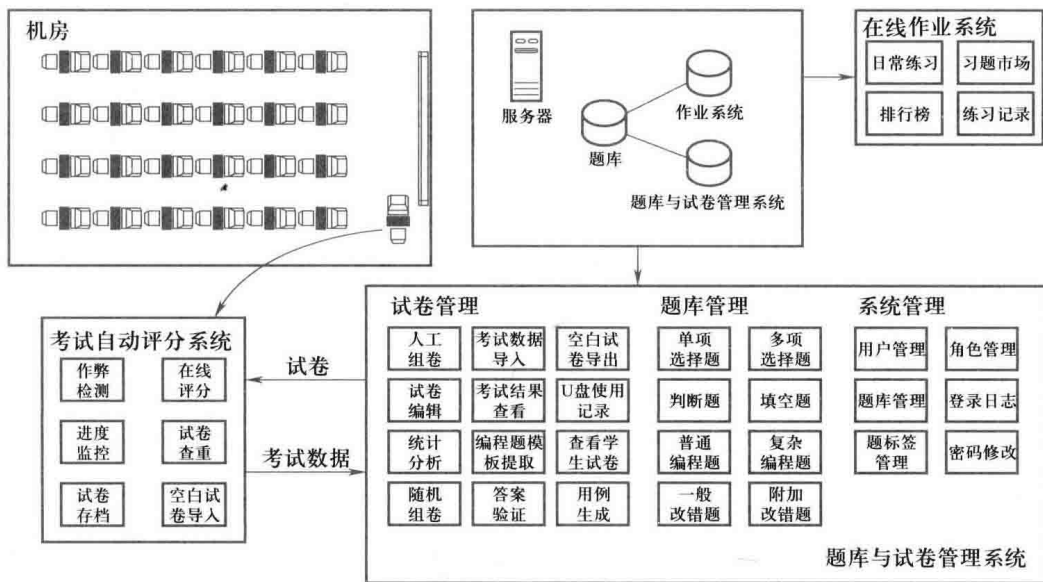
微视频  
Code::Blocks  
的安装过程



微视频  
在 Code::  
Blocks 中新  
建项目



微视频  
在 Code::  
Blocks 中调  
试程序



微视频  
试卷与题库  
管理系统



微视频  
编程题考试自  
动评分系统



微视频  
在线作业系统  
教师用户操作  
演示



微视频  
在线作业系统  
学生用户操作  
演示

各种教学资源之间的关系及介绍

本书由苏小红主编,第1章由孙志岗执笔,第2章由王宇颖执笔,第3~13章由苏小红执笔,第14章由赵玲玲执笔。王甜甜、张彦航、车万翔、孙承杰、张羽、郑贵滨、赵巍、傅忠传、张卫、郭萍、温东新、侯俊英、李希然、单丽莉、刘秉权、袁永峰、叶麟、秦兵、李秀坤、孙大烈、刘旭东、张洪志、王庆北、郝惠馨、李漾、娄久、郭勇等参与了本书的书稿校对工作。北京工业大学的蒋宗礼教授和国防科技大学的徐锡山教授在百忙之中仔细审阅了全部初稿,并提出了许多宝贵的意见和建议。在此对他们的辛勤付出表示衷心的感谢。

因编者水平有限,书中错误在所难免,欢迎读者给作者发送邮件或在网站上留言,对教材提出意见和建议。我们会在每次重印时及时予以更正,读者也可随时从面向读者的教材网站上下载教材的最新勘误表。

# “C 语言程序设计”混合式教学 32 学时分配建议



# 目 录

<b>第1章 为什么要学习编程</b> .....	1	3.2 宏常量与宏替换 .....	30
1.1 学习编程的热潮 .....	1	3.3 const 常量 .....	32
1.1.1 席卷全球的“编程一小时” .....	1	3.4 自动类型转换与强制类型转换 运算符 .....	33
1.1.2 资本汹涌的少儿编程 .....	2	3.5 常用的标准数学函数 .....	35
1.1.3 “再不学编程就晚了” .....	2	3.6 本章知识点小结 .....	37
1.2 为什么要学编程 .....	3	3.7 本章常见错误小结 .....	37
1.3 什么是“编程” .....	5	习题3 .....	38
1.4 怎么学编程 .....	6	<b>第4章 键盘输入和屏幕输出</b> .....	40
1.4.1 一切都是计算 .....	6	4.1 单个字符的输入/输出 .....	40
1.4.2 学编程不是学语法 .....	6	4.1.1 字符常量 .....	40
1.4.3 动手,动手,再动手 .....	7	4.1.2 字符的输入/输出 .....	41
1.5 本章小结 .....	7	4.2 数据的格式化屏幕输出 .....	43
习题1 .....	7	4.3 数据的格式化键盘输入 .....	46
<b>第2章 基本数据类型</b> .....	9	4.4 本章扩充内容 .....	50
2.1 常量与变量 .....	9	4.4.1 用%c输入字符时存在的 问题 .....	50
2.1.1 常量 .....	9	4.4.2 %c格式符存在问题的解决 方法 .....	52
2.1.2 变量 .....	11	4.5 本章知识点小结 .....	54
2.2 简单的屏幕输出 .....	13	4.6 本章常见错误小结 .....	54
2.3 数据类型 .....	14	习题4 .....	55
2.4 如何计算变量或数据类型 所占内存空间的大小 .....	16	<b>第5章 选择控制结构</b> .....	58
2.5 变量的赋值和赋值运算符 .....	18	5.1 生活中与计算机中的问题求解 方法 .....	58
2.6 本章扩充内容 .....	19	5.2 算法的概念及其描述方法 .....	59
2.7 本章知识点小结 .....	22	5.2.1 算法的概念 .....	59
2.8 本章常见错误小结 .....	22	5.2.2 算法的描述方法 .....	60
习题2 .....	23	5.3 关系运算符与关系表达式 .....	61
<b>第3章 基本算术运算</b> .....	24	5.4 用于单分支控制的条件语句 .....	63
3.1 C运算符和表达式 .....	24	5.5 用于双分支控制的条件语句 .....	64
3.1.1 算术运算符和表达式 .....	24		
3.1.2 复合的赋值运算符 .....	27		
3.1.3 增1和减1运算符 .....	28		

5.6	条件运算符和条件表达式 .....	65	7.3.2	函数原型 .....	148
5.7	用于多分支控制的条件语句 .....	65	7.3.3	函数封装与防御性程序 设计 .....	150
5.8	用于多路选择的 switch 语句 .....	71	7.3.4	函数设计的基本原则 .....	157
5.9	逻辑运算符和逻辑表达式 .....	75	7.4	函数的递归调用和递归函数 .....	158
5.10	本章扩充内容 .....	78	7.5	变量的作用域和生存期 .....	161
5.10.1	程序测试 .....	78	7.5.1	变量的作用域 .....	161
5.10.2	对输入非法字符的检查与 处理 .....	87	7.5.2	变量的生存期 .....	166
5.10.3	位运算符 .....	90	7.6	模块化程序设计 .....	170
5.11	本章知识点小结 .....	93	7.6.1	模块分解的基本原则 .....	170
5.12	本章常见错误小结 .....	93	7.6.2	自顶向下、逐步求精 .....	171
习题 5	.....	95	7.6.3	模块化程序设计与多文件 编程实例 .....	172
<b>第 6 章</b>	<b>循环控制结构</b> .....	98	7.6.4	条件编译 .....	180
6.1	循环控制结构与循环语句 .....	98	7.7	本章扩充内容——代码风格 .....	180
6.2	计数控制的循环 .....	107	7.8	本章知识点小结 .....	183
6.3	嵌套循环 .....	110	7.9	本章常见错误小结 .....	183
6.4	条件控制的循环 .....	113	习题 7	.....	184
6.5	流程的转移控制 .....	124	<b>第 8 章</b>	<b>数组和算法基础</b> .....	189
6.5.1	goto 语句 .....	124	8.1	一维数组的定义和初始化 .....	189
6.5.2	break 语句 .....	124	8.2	二维数组的定义和初始化 .....	193
6.5.3	continue 语句 .....	126	8.3	向函数传递一维数组 .....	197
6.5.4	穷举法编程实例 .....	128	8.4	排序和查找 .....	203
6.6	本章扩充内容 .....	131	8.5	向函数传递二维数组 .....	215
6.6.1	结构化程序设计的核心 思想 .....	131	8.6	本章知识点小结 .....	220
6.6.2	常用的程序调试与排错 方法 .....	132	8.7	本章常见错误小结 .....	220
6.6.3	类型溢出 .....	133	习题 8	.....	221
6.7	本章知识点小结 .....	135	<b>第 9 章</b>	<b>指针</b> .....	227
6.8	本章常见错误小结 .....	135	9.1	变量的内存地址 .....	227
习题 6	.....	136	9.2	指针变量的定义和初始化 .....	228
<b>第 7 章</b>	<b>函数与模块化程序设计</b> .....	143	9.3	间接寻址运算符 .....	232
7.1	分而治之与信息隐藏 .....	143	9.4	按值调用与模拟按引用调用 .....	234
7.2	函数的定义 .....	144	9.5	用指针变量作函数参数的程序 实例 .....	240
7.2.1	函数的分类 .....	144	9.6	函数指针及其应用 .....	243
7.2.2	函数的定义 .....	145	9.7	本章知识点小结 .....	251
7.3	向函数传递值和从函数返回值 .....	146	9.8	本章常见错误小结 .....	251
7.3.1	函数调用 .....	146	习题 9	.....	252

<b>第 10 章 字符串</b> .....	255	12.1 从基本数据类型到抽象数据类型	331
10.1 字符串字面量 .....	255	类型 .....	331
10.2 字符串的存储 .....	255	12.2 结构体的定义 .....	332
10.3 字符指针 .....	257	12.2.1 为什么要定义结构体类型 .....	332
10.4 字符串的访问和输入/输出 .....	258	12.2.2 结构体变量的定义 .....	334
10.4.1 如何访问字符串中的单个字符 .....	258	12.2.3 用 typedef 定义数据类型 .....	336
10.4.2 字符串的输入/输出 .....	258	12.2.4 结构体变量的初始化 .....	337
10.5 字符串处理函数 .....	263	12.2.5 嵌套的结构体 .....	338
10.6 向函数传递字符串 .....	266	12.2.6 结构体变量的引用 .....	339
10.7 从函数返回字符串指针 .....	270	12.2.7 结构体所占内存的字节数 .....	343
10.8 本章扩充内容 .....	273	12.3 结构体数组的定义和初始化 .....	345
10.8.1 const 类型限定符 .....	273	12.3.1 结构体数组的定义 .....	345
10.8.2 字符处理函数 .....	274	12.3.2 结构体数组的初始化 .....	345
10.8.3 数值字符串向数值的转换 .....	279	12.4 结构体指针的定义和初始化 .....	347
10.9 本章知识点小结 .....	280	12.4.1 指向结构体变量的指针 .....	347
10.10 本章常见错误小结 .....	281	12.4.2 指向结构体数组的指针 .....	348
习题 10 .....	282	12.5 向函数传递结构体 .....	348
<b>第 11 章 指针和数组</b> .....	285	12.6 共用体 .....	355
11.1 指针和一维数组间的关系 .....	285	12.7 枚举数据类型 .....	359
11.2 指针和二维数组间的关系 .....	293	12.8 动态数据结构——单向链表 .....	360
11.3 指针数组及其应用 .....	300	12.8.1 问题的提出 .....	360
11.3.1 指针数组用于表示多个字符串 .....	300	12.8.2 链表的定义 .....	361
11.3.2 指针数组用于表示命令行参数 .....	306	12.8.3 单向链表的建立 .....	362
11.4 动态数组 .....	308	12.8.4 单向链表的删除操作 .....	366
11.4.1 C 程序的内存映像 .....	308	12.8.5 单向链表的插入操作 .....	367
11.4.2 动态内存分配函数 .....	309	12.9 本章扩充内容 .....	370
11.4.3 长度可变的一维动态数组 .....	311	12.9.1 栈和队列 .....	370
11.4.4 长度可变的二维动态数组 .....	313	12.9.2 树和图 .....	382
11.5 本章扩充内容 .....	315	12.9.3 数据的逻辑结构和存储结构 .....	390
11.5.1 常见的内存错误及其对策 .....	315	12.10 本章知识点小结 .....	392
11.5.2 缓冲区溢出攻击 .....	323	12.11 本章常见错误小结 .....	392
11.6 本章知识点小结 .....	326	习题 12 .....	393
11.7 本章常见错误小结 .....	326	<b>第 13 章 文件操作</b> .....	394
习题 11 .....	327	13.1 二进制文件和文本文件 .....	394
<b>第 12 章 结构体和数据结构基础</b> .....	331	13.2 文件的打开和关闭 .....	396
		13.3 按字符读写文件 .....	397
		13.4 按格式读写文件 .....	404
		13.5 按数据块读写文件 .....	410

13.6 本章扩充内容 .....	415	<b>附录 A</b> C 关键字 .....	439
13.6.1 文件的随机读写 .....	415	<b>附录 B</b> GCC 中基本数据类型的取值 范围 .....	440
13.6.2 标准输入/输出重定向 .....	418	<b>附录 C</b> C 运算符的优先级与结 合性 .....	441
13.7 本章知识点小结 .....	420	<b>附录 D</b> 常用字符与 ASCII 码值对 照表 .....	442
13.8 本章常见错误小结 .....	420	<b>附录 E</b> 常用的 ANSI C 标准库 函数 .....	445
习题 13 .....	420	<b>参考文献</b> .....	453
<b>第 14 章 简单的游戏设计</b> .....	422		
14.1 动画的基本原理 .....	422		
14.2 迷宫游戏 .....	423		
14.3 Flappy bird 游戏 .....	430		
14.4 本章知识点小结 .....	437		
14.5 本章常见错误小结 .....	437		
习题 14 .....	437		

# 第 1 章 为什么要学习编程



## 内容导读

本章力图从一个客观的角度讲述学习编程的意义,编程已不只是一项专业技能,而是与信息世界对话的工具。身处信息世界的人都应该掌握这个工具。放心,它并没有传说中的那么难。

## 1.1 学习编程的热潮

每个时代都有代表性的酷炫技能。比如原始社会是打猎,战争年代是武术,贫乏年代是烧得一手好菜。现在是信息时代、互联网时代,最酷炫的技能是什么呢? PS? 做 UP 主? 游戏高手? 抖音红人? 这些看上去是很酷,也够炫,但其实都是使用别人制作的 App 或网站,在别人设定的规则中放飞自我。真正酷炫的,是制定这些让人痴迷、甚至疯狂的规则的人。其背后的核心技能就是编程。

编程是信息时代的魔法,不会编程的人,只能是“麻瓜”。《哈利·波特》的世界里,绝大多数麻瓜都不知道魔法的存在,这是幸福的。但我们,都知道编程技术的存在,知道那些能神乎其神施展魔法的人,这让我们怎能安心做个麻瓜呢? 更何况,学习编程的旋风已经刮起来了。

### 1.1.1 席卷全球的“编程一小时”

2013 年,公益网站 code.org 发起了一个在每年 12 月持续一周的活动,名为“Hour of Code”,中文译为“编程一小时”。官方是这样介绍的:

“编程一小时是一个介绍计算机科学一小时的活动,旨在揭秘编程并说明它是任何人都可以学习的基础知识。”

这个活动得到了时任美国总统巴拉克·奥巴马,以及科技巨头马克·扎克伯格(Facebook 创始人、CEO)、比尔·盖茨(Microsoft 创始人、董事长)等人的支持。他们特意拍摄视频号召人们参加这个活动。

奥巴马说:“不要只是玩手机,为它编程吧! ……无论你是男生还是女生,无论住在城市还是乡村,都要去试试。”

比尔·盖茨甚至还亲自出镜为初学者制作了教学视频,并勉励大家说:“在今天,编程已经容易了太多。”

2014 年,活动继续,奥巴马不只继续拍摄视频号召美国人民参加,而且还亲身参与了在白宫

举办的一场活动。

美国的热潮迅速影响全球。时至今日,据 code.org 官网统计,已经有 140 多个国家,超过千万人参加了编程一小时。参加者中年龄最小的只有 4 岁,最大的达到 104 岁。

截至 2017 年年底,在中国,该活动已举办了 161 场。与美国的 47 826 场相比,这个数字实在是小得可怜。但没有人怀疑,中国人也都需要学习编程这一命题。至少风险投资的人们在用真金白银表明态度。

### 1.1.2 资本汹涌的少儿编程

读者可能对资本运作并不了解,所以这里先简单科普下。风险投资之所以叫“风险”,是因为他们的投资,有获得几十、上百甚至上千倍回报的可能,同时,血本无归的可能性也很大。高风险和高回报并存,因为他们投资的主要是初创型公司。

初创型公司一般都没有盈利,甚至连获取收入的模式都没有,只有一个美好的愿景,俗称“烧钱”。被烧掉的钱就来自风险投资。如果烧成功了,公司做大了,上市了,那风险投资的回报会非常可观。比如 1999 年,日本软银的孙正义给阿里巴巴投资了 2 000 万美元,占了一定的股份。2014 年阿里巴巴在美国上市时,孙正义所持股份价值 580 亿美元,14 年时间翻了接近 3 000 倍。2018 年中,阿里巴巴股价比上市时又翻了一番,所以孙正义的股份已价值千亿美元,成为了日本首富。但成功的公司其实是罕有的,创业成功率小于 1%,只不过我们看到的,只有成功者而已。所以更多情况下,投资人的钱是被烧得连灰都不剩的。

投资人为什么要冒这么大的风险? 因为只要投中一个阿里巴巴,回报就远超对几百个不成功公司的投资啊。所以投资的基本逻辑就是找“风口”,在有更好未来的领域投更有可能成功的公司。风口总是在变,比如我们都感受过的移动互联网、打车、外卖、共享单车等。那汹涌的补贴、红包,都是投资人为了市场份额而洒下的真钱啊。到了 2017—2018 年,风口吹向了“少儿编程”。

顾名思义,少儿编程的意思就是教少年儿童编程。在其爆发的 2017 年,共有 19 笔总值超 2 亿人民币的融资。但这只是开始。2018 年仅上半年,就有共 18 笔总值近 10 亿人民币的融资。资本汹涌而来,不为别的,就是因为他们相信,会有越来越多的小学生、中学生要接受编程教育,这里有无限商机。尤其浙江省率先全国将信息技术纳入了高考的选考科目范围,更让从业者信心大增。那为什么中小學生都要学习编程呢? 可能是因为孩子家长在为自己不懂编程而恐慌吧。

### 1.1.3 “再不学编程就晚了”

在 2016 年之前,网上各大在线教育机构提供的各类编程基础课程,学习者是以在校大学生为主,尤其是即将找工作的大三、大四学生。这很好理解,IT 行业薪资高、发展空间大,非计算机专业的希望能转行,计算机专业的希望能更有竞争力,所以会选择在网上学更接近实战的课程。但 2016 年之后,成年人比例就开始逐渐上升,其中不乏已有稳定工作甚至功成名就的人士。他

们为什么要学编程呢？有一位学员说了这样一句话：

“再不学编程就晚了！”

为什么再不学编程就晚了？他恐慌的是什么呢？

## 1.2 为什么要学编程

人人学编程 ≠ 人人都是程序员。

很多人误解“人人学编程”，以为其目的是补充程序员的人力空缺，其实不是。程序员不是人人都想当的，更不是人人都能做的。那为什么要人人学编程？为什么 2016 年突然很多成年人开始学编程，随后少儿编程市场就火爆了？那一年发生了什么？

2016 年 3 月，Google 开发的人工智能围棋软件(AlphaGo)挑战世界冠军、韩国著名围棋棋手李世石，最终以 4:1 获胜。其后的所有人机博弈中，包括 2017 年对弈排名第一的人类棋手柯洁，AlphaGo 都再未败过一局，独孤求败样地退役了。围棋，是人类顶级的脑力竞赛，却被人工智能打败了……人工智能会统治世界吗？人类将何去何从？

此时，我们最能直接想到的答案是：人工智能是用程序写出来的，只有掌握了编程能力，才能凌驾于人工智能之上。

基于此，很多人产生了自己或让孩子学编程的念头。听起来好像很有逻辑，但其实，这个论断并不正确。人工智能也可以写代码，这可怎么办呀？在当下，恐慌于人工智能，是大可不必的。但当下可以看到的是，软件已经改变了世界，改变了我们做事的方式，而人工智能可以加剧改变。如果我们能主导、参与、适应这些变化，那么就能更好地利用这些变化。在这个过程中，需要具备一种思维能力，这种能力叫“计算思维”。编程是学习这种能力的最佳途径。

美国卡内基·梅隆大学计算机科学系前系主任周以真教授在 2006 年发表了一篇著名的文章——《计算思维(Computational Thinking)》。文中谈到“计算机科学的教授应当为大学新生开一门称为‘怎么像计算机科学家一样思维’的课，面向非专业的，而不仅仅是计算机科学专业的学生”，这是因为“机器学习已经改变了统计学。……计算生物学正在改变着生物学家的思考方式。类似地，计算博弈理论正改变着经济学家的思考方式，纳米计算改变着化学家的思考方式，量子计算改变着物理学家的思考方式”，所以“计算思维代表着一种普遍的认识和一类普适的技能，每一个人，不仅仅是计算机科学家，都应热心于它的学习和运用”。

我很赞同她的观点，并尝试用更通俗的语言来解释为什么人人都应该掌握计算思维。我们所处的时代被称作“信息时代”，计算机是信息处理的核心。商业、农业、工业、教育等都被计算机技术推动着进步。如果能知道计算机是怎么做到这一切的，也就是知道这里蕴含的计算思维是什么，那么可能给个人所在行业的发展带来帮助。否则，就只能浮于表面地应用现成软件而已。比方说，人类一直在用推理来解决很多需要思考的问题，如下棋。所以看到会下棋的程序，很多人都认为这是计算机在思考，很神奇。实际上，计算机从来都不会思考。

1997 年打败国际象棋冠军卡斯帕罗夫的 IBM 深蓝电脑，使用的核心方法叫做“搜索”，就是