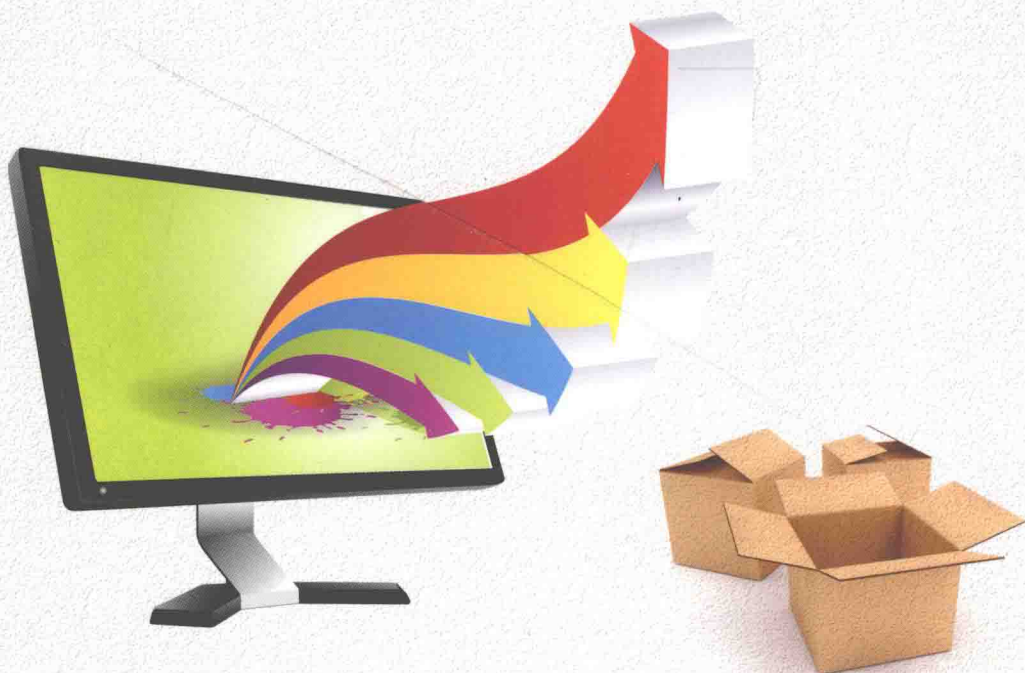


高等学校计算机类课程应用型人才培养规划教材

C语言程序设计

C Programming

李晓东 庞岩梅 娄嘉鹏 谭思哲 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高等学校计算机类课程应用型人才培养规划教材

C 语言程序设计

李晓东 庞岩梅 娄嘉鹏 谭思哲 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书由“入门篇、初级篇、中级篇、提高篇、综合篇”5部分组成：入门篇是本书的先导部分，让读者了解计算的机器、高级语言、C语言的历史和特点等；初级篇是让读者掌握简单问题的C语言编程，学会把C语言和已有的知识和想法建立关联，对于简单问题能由想法写出代码；中级篇是让读者掌握复杂问题的C语言编程，学会通过算法解决问题，对于复杂的问题能由想法获得算法，再由算法写出代码；提高篇是让读者掌握如何写出高质量的代码，前面各篇立足于能编出代码解决问题即可，本篇则让读者学会编写高质量的代码，在C语言编程上产生由量到质的提升；综合篇是让读者掌握应用领域编程及实用工具的开发过程。本书中以星号(*)开头的节为知识点的详细讲解，供读者编程实践时参考。

本书适合为应用型高等学校电气信息类专业学习C语言程序设计的教材，高职高专院校可酌情选为相关专业的教材，本书也适于作为C语言爱好者的自学用书。

图书在版编目(CIP)数据

C语言程序设计/李晓东等编著. — 北京：中国铁道出版社，2012.1

高等学校计算机类课程应用型人才培养规划教材

ISBN 978-7-113-14099-1

I. ①C… II. ①李… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第278874号

书 名：C语言程序设计

作 者：李晓东 庞岩梅 姜嘉鹏 谭思哲 编著

策 划：严晓舟 焦金生

读者热线：400-668-0820

责任编辑：周海燕 冯彩茹

封面设计：付 巍

封面制作：白 雪

责任印制：李 佳

出版发行：中国铁道出版社(100054,北京市西城区右安门西街8号)

网 址：<http://www.edusources.net>

印 刷：航远印刷有限公司

版 次：2012年1月第1版 2012年1月第1次印刷

开 本：787mm×1092mm 1/16 印张：18 字数：423千

印 数：1~3 000册

书 号：ISBN 978-7-113-14099-1

定 价：30.00元

版权所有 侵权必究

凡购买铁道版图书，如有印制质量问题，请与本社教材图书营销部联系调换。电话：(010) 63550836

打击盗版举报电话：(010) 63549504

编 审 委 员 会

主 任: 何新贵 教授、中国工程院院士 (北京大学)

副主任: (按姓氏笔画排序)

刘瑞挺	教授	(南开大学)
罗晓沛	教授	(中国科学院研究生院)
岳丽华	教授	(中国科学技术大学)
周广声	教授	(上海海事大学)
郑人杰	教授	(清华大学)
徐洁磐	教授	(南京大学)
唐世渭	教授	(北京大学)

委 员: (按姓氏笔画排序)

王 浩	教授	(合肥工业大学)
王晓峰	教授	(上海海事大学)
史九林	教授	(南京大学)
白晓颖	教授	(清华大学)
刘 强	副教授	(清华大学)
许 勇	教授	(安徽师范大学)
孙志挥	教授	(东南大学)
李龙澍	教授	(安徽大学)
李银胜	副教授	(复旦大学)
李盛恩	教授	(山东建筑大学)
李敬兆	教授	(安徽理工大学)
杨吉江	教授	(清华大学)
何守才	教授	(上海第二工业大学)
余 粟	副教授	(上海工程技术大学)
张 莉	教授	(北京航空航天大学)
张 瑜	教授	(上海工程技术大学)
张燕平	教授	(安徽大学)
陈世福	教授	(南京大学)
陈涵生	研究员	(上海华东计算技术研究所)
迟洪钦	副教授	(上海师范大学)
林钧海	教授	(南京航空航天大学)
金志权	教授	(南京大学)
周鸣争	教授	(安徽工程科技学院)
周根林	教授级高级工程师	(中电集团)
胡学钢	教授	(合肥工业大学)
姜同强	教授	(北京工商大学)
徐永森	教授	(南京大学)
殷人昆	教授	(清华大学)
郭学俊	教授	(河海大学)
黄上腾	教授	(上海交通大学)
董继润	教授	(山东大学)
蔡瑞英	教授	(南京工业大学)

当前，世界格局深刻变化，科技进步日新月异，人才竞争日趋激烈。我国经济建设、政治建设、文化建设、社会建设以及生态文明建设全面推进，工业化、信息化、城镇化和国际化深入发展，人口、资源、环境压力日益加大，调整经济结构、转变发展方式的要求更加迫切。国际金融危机进一步凸显了提高国民素质、培养创新人才的重要性和紧迫性。我国未来发展关键靠人才，根本在教育。

高等教育承担着培养高级专门人才、发展科学技术与文化、促进现代化建设的重大任务。近年来，我国的高等教育获得了前所未有的发展，大学数量从1950年的220余所已上升到2008年的2200余所。但目前诸如学生适应社会以及就业和创业能力不强，创新型、实用型、复合型人才紧缺等高等教育与社会经济发展不相适应的问题越来越凸显。2010年7月发布的《国家中长期教育改革和发展规划纲要（2010—2020年）》提出了高等教育要“建立动态调整机制，不断优化高等教育结构，重点扩大应用型、复合型、技能型人才培养规模”的要求。因此，新一轮高等教育类型结构调整成为必然，许多高校特别是地方本科院校面临转型和准确定位的问题。这些高校立足于自身发展和社会需要，选择了应用型发展道路。应用型本科教育虽早已存在，但近几年才开始大力发展，并根据社会对人才的需求，扩充了新的教育理念，现已成为我国高等教育的一支重要力量。发展应用型本科教育，也成为中国高等教育改革与发展的重要方向。

应用型本科教育既不同于传统的研究型本科教育，又区别于高职高专教育。研究型本科培养的人才将承担国家基础型、原创型和前瞻型的科学研究，它应培养理论型、学术型和创新型的研究人才。高职高专教育培养的是面向具体行业岗位的高素质、技能型人才，通俗地说，就是高级技术“蓝领”。而应用型本科培养的是面向生产第一线的本科层次的应用型人才。由于长期受“精英”教育理念的支配，脱离实际、盲目攀比，高等教育普遍存在重视理论型和学术型人才培养的偏向，忽视或轻视应用型、实践型人才的培养。在教学内容和教学方法上过多地强调理论教育、学术教育而忽视实践能力的培养，造成我国“学术型”人才相对过剩，而应用型人才严重不足的被动局面。

应用型本科教育不是低层次的高等教育，而是高等教育大众化阶段的一种新型教育层次。计算机应用型本科的培养目标是：面向现代社会，培养掌握计算机学科领域的软硬件专业知识和专业技术，在生产、建设、管理、生活服务等第一线岗位，直接从事计算机应用系统的分析、设计、开发和维护等实际工作，维持生产、生活正常运转的应用型本科人才。计算机应用型本科人才有较强的技术思维能力和技术应用能力，是现代计算机软、硬件技术的应用者、实施者、实现者和组织者。应用型本科教育强调理论知识和实践知识并重，相应地，其教材更强调“用、新、精、适”。所谓“用”，是指教材的“可用性”、“实用性”和“易用性”，即教材内容要反映本学科基本原理、思想、技术和方法在相关现实领域的典型应用，介绍应用的具体环境、条件、方法和效果，培养学生根据现实问题选择合适的科学思想、理论、技术和方法去分析、解决实际问题的能力。所谓“新”，是指教材内容应及时反映本学科的最新发展和最新技术成就，以及这些新知识和新成就在行业、生产、管理、服务等方面的最新应用，从而有效地保证学生“学以致用”。所谓“精”，不是一般意义的“少而精”。事实常常告诉我们“少”与“精”是有矛盾的，数量的减少并不能直接促使质量的提高。而且，“精”又是对“宽与厚”的直接“背叛”。

因此,教材要做到“精”,教材的编写者要在“用”和“新”的基础上对教材的内容进行去伪存真的精练工作,精选学生终身受益的基础知识和基本技能,力求把含金量最高的知识传承给学生。“精”是最难掌握的原则,是对编写者能力和智慧的考验。所谓“适”,是指各部分内容的知识深度、难度和知识量要适合应用型本科的教育层次,适合培养目标的既定方向,适合应用型本科学生的理解程度和接受能力。教材文字叙述应贯彻启发式、深入浅出、理论联系实际、适合教学实践,使学生能够形成对专业知识的整体认识。以上4个方面不是孤立的,而是相互依存的,并具有某种优先顺序。“用”是教材建设的唯一目的和出发点,“用”是“新”、“精”、“适”的最后归宿。“精”是“用”和“新”的进一步升华。“适”是教材与计算机应用型本科培养目标符合度的检验,是教材与计算机应用型本科人才培养规格适应度的检验。

中国铁道出版社同高等学校计算机类课程应用型人才培养规划教材编审委员会经过近两年的前期调研,专门为应用型本科计算机专业学生策划出版了理论深入、内容充实、材料新颖、范围较广、叙述简洁、条理清晰的系列教材。本系列教材在以往教材的基础上大胆创新,在内容编排上努力将理论与实践相结合,尽可能反映计算机专业的最新发展;在内容表达上力求由浅入深、通俗易懂;编写的内容主要包括计算机专业基础课和计算机专业课;在内容和形式体例上力求科学、合理、严密和完整,具有较强的系统性和实用性。

本系列教材是针对应用型本科层次的计算机专业编写的,是作者在教学层次上采纳了众多教学理论和实践的经验及总结,不但适合计算机等专业本科生使用,也可供从事IT行业或有关科学研究工作的人员参考,适合对该新领域感兴趣的读者阅读。

本系列教材出版过程中得到了计算机界很多院士和专家的支持和指导,中国铁道出版社多位编辑为本系列教材的出版做出了很大贡献,在此表示感谢。本系列教材的完成不但依靠了全体作者的共同努力,同时也参考了许多中外有关研究者的文献和著作,在此一并致谢。

应用型本科是一个日新月异的领域,许多问题尚在发展和探讨之中,观点的不同、体系的差异在所难免,本系列教材如有不当之处,恳请专家及读者批评指正。

《高等学校计算机类课程应用型人才培养规划教材》编审委员会

2011年1月

现有的《C 语言程序设计》教科书通常都是围绕知识点进行讲解，结果过分地强调了知识细节，学生记忆了大量的知识细节，学习结束后却不会编程。本书的出发点在于围绕问题的解决讲解 C 语言编程。C 语言是为解决问题而诞生的，它是一种出色的解决问题的工具，并经历了长期的实践检验。把握一个工具最好的途径是去体会它为什么能出色地解决问题。例如，当我们在使用的特殊钳子时体会到它的妙处，马上就会认为好的钳子就应该是这样的。这时，不用记忆，我们也会把握钳子的使用要领，甚至加以发挥。因此，越好的语言越容易掌握，因为它正是用来方便我们解决问题的。本书围绕问题解决讲解 C 语言编程，让学生在问题解决中感性切入并逐步提升，而不是在 C 语言的知识细节中徘徊。

本书具有以下特点：

(1) 框架完整，知识细节不要求必须记忆，但要求掌握如何探索知识细节。

(2) 按照问题解决的需要分多个层次逐步讲解 C 语言编程，依次为：能够由解决问题的想法映射到代码，能够由解决问题的想法到算法再到最终的代码，能够写出高质量的代码，能够进行应用领域编程，以及能够进行简单实用工具的开发。实际应用中大多数的编程属于简单编程，这就是说我们只要学会由解决问题的想法到程序代码的映射，已能够解决大多数的实际问题。

(3) 注意与其他课程的衔接。本书针对学生以后学习数据结构、面向对象编程、嵌入式开发编程、图形界面编程、软件工程等课程进行了铺垫和延伸，有助于学生过渡到后续课程的学习。

(4) 本书在附录中特意提供了针对遇到什么问题应参考书籍的指南。

本书的内容分为入门篇、初级篇、中级篇、提高篇、综合篇，各篇的划分总体体现了入门、简单、复杂、深化、综合的线索。入门篇是本书的先导部分，让学生了解计算的机器、高级语言、C 语言的历史和特点等；初级篇是让学生掌握简单问题的 C 语言编程，学会把 C 语言和已有的知识和想法建立关联，对于简单问题能由想法写出代码；中级篇是让学生掌握复杂问题的 C 语言编程，学会通过算法解决问题，对于复杂的问题能由想法获得算法，再由算法写出代码；提高篇是让学生掌握如何写出高质量的代码，前面各篇立足于能编出代码解决问题即可，本篇则让学生学会如何编写高质量的代码，在 C 语言编程上产生由量到质的提升；综合篇是让学生掌握如何进行应用领域编程及实用工具的开发过程。

本书的第 1 章和第 3 章以及第 2 章和第 6 章的部分由李晓东编写，第 4 章和第 5 章以及第 2 章的部分由庞岩梅编写，第 7 章~第 9 章由娄嘉鹏编写，第 6 章的部分由谭思哲编写。徐小青、章小莉、张晓昆、刘瑾、谢绒娜在本书的编写过程中提出了许多宝贵的建议，史九林教授审阅了全书，在此一并表示感谢。

由于时间仓促，加之编者水平有限，书中难免存在不足和疏漏之处，恳请广大读者批评指正。

编 者

2011 年 9 月

CONTENTS

目 录

第一篇 入 门 篇

第 1 章 高级语言简介	3
1.1 计算的机器	3
1.2 如何与机器交流	4
1.3 C 语言简介	5
1.3.1 C 语言的历史	5
1.3.2 C 语言的特点	6
本章小结	7
习题	7

第二篇 初 级 篇

第 2 章 基本编程	11
2.1 C 语言实例演示	13
2.2 熟悉 C 语言开发环境	18
2.2.1 C 语言程序编制简单流程	18
2.2.2 开发环境的建立	18
2.2.3 自动生成 Hello World! C 语言代码	19
2.2.4 编译和连接程序	24
2.2.5 运行程序显示 hello world!	25
2.2.6 改变程序的显示内容	25
2.2.7 让非自动生成的 C 语言代码可以直接运行	26
2.3 读懂一个 C 语言程序	26
2.4 编一个程序输出一段文字	28
2.4.1 输出一行字符	28
2.4.2 输出多行字符	28
2.4.3 ASCII 码	29
2.4.4 语句的概念	29
2.4.5 如何调用外部函数	30
*2.4.6 printf() 函数的使用	30
2.5 编一个程序完成密码算法中的回归	32
2.5.1 C 语言代码	32
2.5.2 变量声明及初始化	33
2.5.3 如何在 C 语言中做判断	34
2.5.4 判断的嵌套	35

2 C 语言程序设计

2.5.5	如何输出数据	35
*2.5.6	关键字	36
*2.5.7	用户自定义标识符定义规则	37
*2.5.8	基本数据类型	37
*2.5.9	判断(分支)结构	41
2.6	编一个程序进行求模运算	43
2.6.1	C 语言代码	43
2.6.2	让 C 语言执行运算	43
2.6.3	表达式的概念	46
*2.6.4	表达式的运算规则	46
2.7	编一个程序进行字符串的加密	48
2.7.1	C 语言代码	48
2.7.2	循环的概念	48
2.7.3	循环的嵌套	48
*2.7.4	数组类型	49
*2.7.5	循环结构	49
2.8	编一个程序对键盘输入的字符串进行加密	51
2.8.1	C 语言代码	51
2.8.2	输入数据的方法	51
2.8.3	与用户交互	52
2.8.4	scanf()函数的使用	53
2.9	调试	56
2.9.1	模拟计算机执行	56
2.9.2	常见的简单错误	57
2.10	由想法映射到代码	58
本章小结		62
习题		63

第三篇 中 级 篇

第 3 章	算法设计	67
3.1	编写一个判断任意给定数是否为素数的程序	68
3.1.1	思路 and 步骤	68
3.1.2	C 语言代码	69
3.2	算法的概念	71
3.2.1	什么是算法	71
3.2.2	算法的描述	72
*3.3	算法的结构	75
3.3.1	算法的基本执行结构	75
3.3.2	逐步求精	77
3.4	算法的数据组织	79

3.4.1	数组	80
3.4.2	多维数组	81
3.4.3	结构体	82
3.4.4	指针	83
3.4.5	链表	84
3.4.6	树和图	85
3.4.7	数据类型的扩展机制	85
3.4.8	利用数据组织获得好的算法	86
3.5	典型算法整理	86
3.5.1	求累加和	86
3.5.2	求累乘积	86
3.5.3	求阶乘	87
3.5.4	查找	87
3.5.5	排序	88
3.5.6	进制转换	91
3.5.7	求最大公约数及最小公倍数	92
3.5.8	数值求解	94
	本章小结	95
	习题	95
第 4 章	算法实现	97
4.1	常用的数据组织形式	98
4.1.1	数据类型扩展	98
4.1.2	数组及其使用	98
4.1.3	结构体及其使用	106
4.1.4	指针及其使用	110
4.1.5	链表及其使用	113
4.1.6	枚举类型	118
*4.1.7	复合数据类型小结	119
4.2	由算法到代码	125
4.2.1	顺序结构	125
4.2.2	选择结构	126
4.2.3	循环结构	129
4.3	调试方法	133
4.3.1	结构化调试	133
4.3.2	自底向上调试	133
4.3.3	自顶向下调试	133
4.3.4	VC 环境下的调试机制	133
4.3.5	程序调试的常用技巧和调试中问题的解决	134
	本章小结	134
	习题	134

第 5 章 文件的使用——数据持久存储	136
5.1 文件的分类	137
5.2 文件的处理	137
5.2.1 文件的打开	137
5.2.2 文件的关闭	139
5.2.3 文件的读取	140
5.2.4 文件的写入	141
5.2.5 文件结束的判断	144
5.2.6 文件的其他操作	145
5.3 使用配置文件	146
*5.4 文件访问的常用函数	148
本章小节	149
习题	149

第四篇 提 高 篇

第 6 章 代码质量	153
6.1 模块化	154
6.1.1 变量的作用范围	154
6.1.2 服务的概念与函数的使用	156
6.1.3 多个文件模块	158
*6.1.4 局部变量、全局变量和静态变量	159
*6.1.5 C 语言函数的使用	164
6.2 编程风格	176
6.2.1 代码格式	176
6.2.2 注释的使用	178
6.2.3 宏的使用	179
6.2.4 其他	180
*6.2.5 宏的具体用法	182
6.3 提高代码可靠性	186
6.3.1 常量的定义和使用	186
6.3.2 异常情况的判断和处理	187
6.3.3 代码测试方法	188
*6.3.4 常量的使用	189
6.4 友好的界面	190
6.4.1 友好界面的要素	190
6.4.2 菜单的使用	190
6.4.3 命令行界面的使用	191
6.5 性能提高	192
6.5.1 性能的分析	192
6.5.2 循环次数的估计	193

6.5.3 性能的提高方法	194
*6.5.4 break、continue、return 和 exit 的灵活使用	194
6.5.5 内存资源的动态申请和释放	196
*6.5.6 内存资源管理函数的使用	198
本章小结	200
习题	201

第五篇 综 合 篇

第 7 章 应用领域编程	205
7.1 应用接口的调用	206
7.2 网络开发包的调用实例	216
7.3 图形应用代码编写	221
本章小结	228
习题	228
第 8 章 深入 C 语言的应用	229
8.1 C 语言的标准化过程	230
8.2 面向对象思想到 C++	231
8.3 图形用户界面	235
8.4 嵌入式环境的 C 语言	243
8.5 混合编程	245
本章小结	246
习题	246
第 9 章 应用程序的开发	247
9.1 代码与应用程序的区别	247
9.2 软件过程	248
9.3 一个应用程序的开发实例	249
本章小结	256
习题	256
附录	257
附录 A ASCII 码表	257
附录 B C 语言关键字	258
附录 C C 语言常用语法表	259
附录 D C 语言常用函数表	265
附录 E 编程参考书推荐	269
附录 F C 语言常见调试错误	270
参考文献	271

第一篇

入门篇

本篇讲述了计算机高级语言的作用及 C 语言的历史和特点。要求学生了解计算的机器、高级语言、C 语言的历史和特点，理解计算机语言的作用。

第1章 高级语言简介

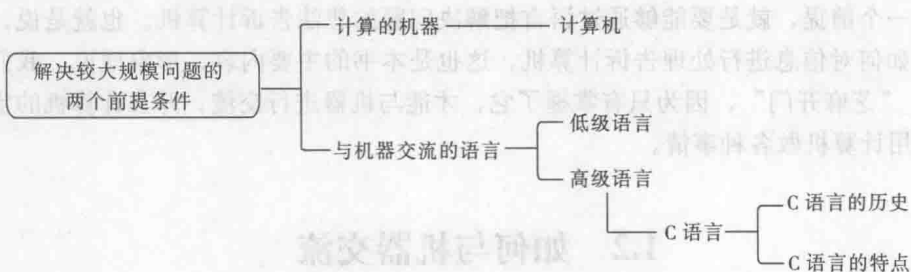
本章导读

本章重点介绍高级语言的一些基本知识,包括计算机高级语言的作用,C语言的历史和特点等。

内容要点:

- 计算的机器;
- 如何与机器交流;
- C语言的历史与特点。

内容结构



学习目标

本章内容只要求做认识性了解,通过学习,应该能:

- 初步建立对编程语言的全面认识;
- 熟悉C语言的历史和特点;
- 为今后的深入学习和研究打下基础。

1.1 计算的机器

设想有一天,你面对一个“挖土”的任务(见图1-1),你会怎么做?

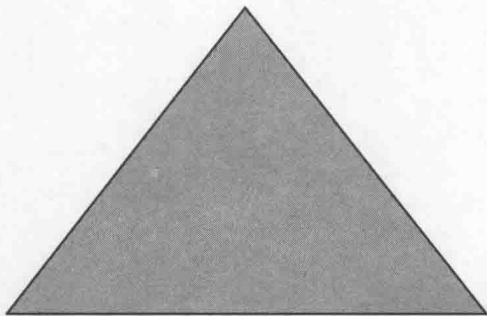


图 1-1 “挖土”的任务

办法有两个：①找来铁锹，一锹一锹地挖，直到挖完为止；②找到一个会干活的机器，比如挖土机，告诉它怎么做，让它去完成这个任务。

显然，第 1 种办法是手工的办法，是自己亲自动手一步一步地做完。第 2 种办法则是让机器替我们做。前者仅适用于小规模的问题，对于规模较大的问题，需要用第 2 种方法来解决。

用第 2 种方法解决问题需要有两个前提条件：

(1) 要有能够自动工作的机器，对于挖土的任务则需要挖土机或更高级智能的挖土机器。

(2) 要能够通过语言把你解决问题的想法告诉机器，好让机器完成任务。当然，既可以手把手地告诉机器怎么做，也可以把想法告诉一个翻译，再由翻译具体地告诉机器，然后让机器去做。

对于我们来说，能够为我们自动工作的机器早已发明好了，那就是计算机。计算机能够自动、快速、可靠地帮我们解决问题。这是因为，计算机是一个能够自动处理信息的机器，而绝大多数问题的解决都能归结为信息的处理，信息的处理也称为计算。正如挖土机是用来挖土的，对于挖土任务就可以让挖土机去做一样，我们可以把计算的任务交给计算机去做。

还剩下一个前提，就是要能够通过语言把解决问题的想法告诉计算机。也就是说，要能够通过语言把如何对信息进行处理告诉计算机，这也是本书的主要内容，形象地说，我们要学的是 4 个字：“芝麻开门”，因为只有掌握了它，才能与机器进行交流，开启计算机的宝藏，否则就不能利用计算机做各种事情。

1.2 如何与机器交流

我们需要通过机器语言告诉计算机如何处理信息，用计算机能直接明了的语言让它工作，前提是需要事先想好对信息处理的每一个具体的步骤。当然，这样做很累。还有另一种办法，就是请一个专业翻译，我们把信息处理的宏观步骤告诉翻译，由翻译把它转换成一个个具体的步骤让计算机去执行。

因此，我们利用计算机解决问题的同时，也希望有一个好语言。有的语言表达能力强，有的语言则差一些。对于语言，还有一个就是人驾驭的问题。同样是一种语言，比如汉语，要表达同样的一个意思，有的人表达得非常简练，有的人则要说上半天，甚至还可能说不清楚。

例如有的人用简单的语言 BASIC 也能编出非常好的程序，有的人用非常高级的语言 Java 也可能编出非常差的程序。这就需要多编程，在编程实践中多领会语言的妙用，多领会编程的真谛。

1.3 C 语言简介

1.3.1 C 语言的历史

C 语言的诞生是和操作系统分不开的。在 C 语言之前已经有高级语言了，但是这些高级语言并不适合操作系统，因此操作系统只好使用汇编语言来编写。我们知道，操作系统通常要求有强大的功能，而用汇编语言来编写也必然是十分费时和费力的。

为此，有人把当时的 CPL 高级语言改造成 BCPL 语言，后来又改造成 B 语言。UNIX 的创始人之一用 B 语言编出了 UNIX 操作系统，在此基础上，UNIX 的另一个创始人发明了 C 语言，并用 C 语言改写了 UNIX 操作系统。由此，UNIX 开始迅速走向普及，而 C 语言也成为广为流行的高级语言。

和其他事物一样，C 语言的流行带来了另一个问题——C 语言的标准化问题。大家为 C 语言添枝加叶的同时，也不可避免地出现了多种不同的 C 语言，导致它们的代码相互之间不能兼容。于是，标准化时代应运而生。任何事物都经历诞生、成长、壮大的过程，标准化是进行规范的过程，也是一个事物由壮大走向成熟的过程。1983 年，美国国家标准学会（ANSI）根据 C 语言问世以来各种版本所做的发展和扩充，公布了第一个 C 语言标准草案。随后于 1989 年公布了完整的 C 语言标准（ANSI C-C89）。虽然以后 C 语言的标准化工作并没有停止，但目前流行的 C 语言编译系统大多是以 C89 为基础开发的。

目前，最流行的 C 语言版本有以下几种：

- Microsoft C 或称 MS C。
- Borland Turbo C 或称 Turbo C。
- AT&T C。
- gcc。

这些 C 语言版本不仅实现了 ANSI C 标准，而且在此基础上还做了一些扩充，使之更加方便、完美。

C 语言的创始人是 Dennis M. Ritchie（见图 1-2，也是 UNIX 的创始人之一）：

C 语言的发展简史如下：

- 1963 年，剑桥大学将 ALGOL 60 语言（也称为 A 语言）发展成为 CPL（Combined Programming Language）语言。
- 1967 年，剑桥大学的 Martin Richards 对 CPL 语言进行了简化，于是产生了 BCPL 语言。
- 1970 年，美国贝尔实验室（AT&T）的 Ken Thompson 将 BCPL 进行了修改，并为它起了一个有趣的名字“B 语言”，意思是提取 CPL 的精华，并用 B 语言编写了第一个 UNIX 操作系统。
- 1973 年，美国贝尔实验室的 D.M. RITCHIE 在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二字母母作为这种语言的名字，这就是 C 语言。
- 为了使 UNIX 操作系统推广使用，1977 年 Dennis M. Ritchie 发表了不依赖于具体机器系统的 C 语言程序编写“可移植的 C 语言编译程序”。

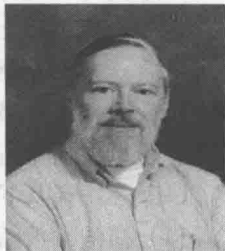


图 1-2 Dennis M. Ritchie