

PEARSON

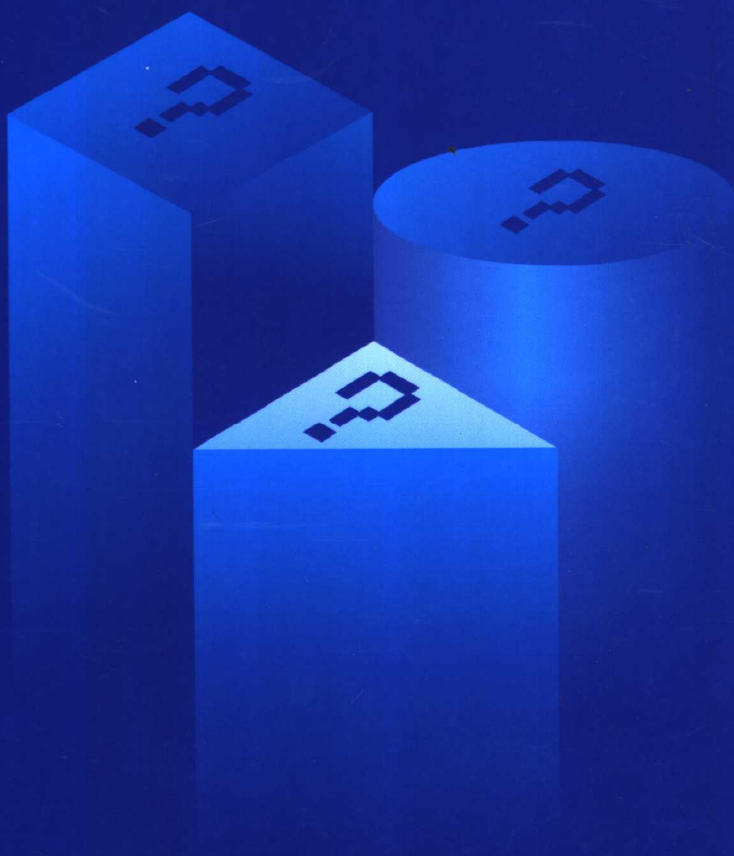
Prentice
Hall

 国外经典教材

问题求解与编程概念 (第6版)

Problem Solving and Programming Concepts (6th Edition)

(美) Maureen Sprankle 著
张晓明 邓劲生 译



清华大学出版社

国外经典教材

问题求解与编程概念 (第 6 版)

(美) Maureen Sprankle 著
张晓明 译
邓劲生

清华大学出版社
北 京

内 容 简 介

本书深入浅出地阐述了如何利用计算机来求解问题的一些思想和方法。全书分为6个部分,另有6个附录。本书全面讲解了解题方法和编程概念的各个方面,通过详细的解释和大量的实例逐步展开介绍,从基本的数学函数和操作符到代码、数组、指针和其他数据结构等技术的设计和使用,以及数据库和面向对象程序设计的概念,然后以综合实例阐述应用软件中解题方法的使用。附录介绍了商业中常用到的模板、排序和合并方法。本书适合作为高等学校计算机专业学生的必修或选修课程,对希望提高编程水平的程序员也有很大参考价值。

Simplified Chinese edition copyright © 2003 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Problem Solving and Programming Concepts, 6th Edition by Maureen Sprankle, Copyright © 2003, 2001, 1998, 1995, 1992

EISBN: 0-13-048268-4

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education, Inc.,

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2003-0561

本书封面贴有 Pearson Education (培生教育出版集团)激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

问题求解与编程概念(第6版)/(美)斯普兰克(Sprankle, M.)著;张晓明,邓劲生译. —北京:清华大学出版社, 2003

(国外经典教材)

书名原文: Problem Solving and Programming Concepts (6th Edition)

ISBN 7-302-07329-5

I. 问… II. ①斯… ②张… ③邓… III. 程序设计—教材 IV. TP311.1

中国版本图书馆CIP数据核字(2003)第088128号

出 版 者: 清华大学出版社

地 址: 北京清华大学学研大厦

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

客户服务: 010-62776969

文稿编辑: 车立红

封面设计: 立日新设计公司

印 刷 者: 北京彩艺印刷有限公司

装 订 者: 三河市新茂装订有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 28.75 字数: 768千字

版 次: 2003年11月第1版 2003年11月第1次印刷

书 号: ISBN 7-302-07329-5/TP·5319

印 数: 1~3500

定 价: 55.00元

译者序

在踏入编程这个门槛时，曾经有无数的各种说法在我们耳边萦绕。各种各样或难或易的论断，让我们一再地为之迷惑不解。经常存在的编程语言之争（比如是使用 C++ 语言还是使用 Java 语言），或者编程环境之争（比如是使用 Visual C++ 还是使用 C++ Builder），常常让我们无所适从。面对这种境况，我们不由自主地想起站在两堆同样草料正中央的那头笨驴，为着这无法做出的决定而犹豫再三无从取舍。

这时，如果有人告诉你，其实编程语言都一样，你会有什么感觉？这是很多年前一位德高望重的老教授说过的一句话。当时，我们正为 Pascal 的严格语法而劳心费神，而 C++ 语言正在逐步进入高校，却被指责为过于灵活从而不利于培养编程风格。两者的语法是如此的不同！当时我们的感觉是大为疑惑。

10 年过去了。DOS 以及伴随它的那些编程工具，已经不再被现在进入编程领域的人们所提起。在经历过编程工具一次次的版本更替和环境迁移之后，回头看看各种语言的教材这些年的发展，我们惊讶地发现，其实在解题方法和编程概念方面，还是那些东西。一个某种语言的编程专家，可以迅速地跨越到另一种新的语言或者另一种操作系统平台，他凭的是什么呢？这就是本书所要揭示的内容。

授人以鱼不如授人以渔。掌握一门编程语言从而能编写几十行几百行顺畅运行的代码，在当前 IDE 盛行的年代已经不是难事。但是，要想掌握一种解题方法，却不是每个人都能够轻易做到的。本书能够在短短的一个学期内，将目前编程领域的最新解题方法都让读者有所了解，这是作者的高明之处。

本书面向的读者是计算机编程的初学者。从刚进入编程领域就开始培养一种思想和理念，是现代计算机教育的重大改进。这样才能够让编程人员将眼光不仅仅局限于某几行代码的简洁度、某个算法的巧妙性或者某个界面的亲和力（当然这些也都是应该注意的），而是可以站在更高的角度上，看到更远的编程空间。

在翻译过程中，我们也经历了一次再学习的过程。本书译稿经过多人试读，我们发现任何编程人员在阅读本书后都会有所收获，不但可以从中学习到一些求解问题的方法，同时对于培养求解问题的创造性思维和基本技能也大有裨益。现在我们将本书推荐给广大读者，希望您能从中受益。

本书的主要翻译工作由张晓明和邓劲生协作完成，汪黎、文茜、何定、周建飞、陈刚、李云照、陆平静、祝琳、刘明辉等同志参加了部分翻译、校对、录入和排版工作。此外，杨明军、王英、刘兴昉、李伟洲、方兴、李宝峰、颜亮等同志在阅读初稿的过程中提出了许多宝贵的修改意见，在此一并表示感谢。由于我们的水平有限，译文不确切与理解谬误之处在所难免，谨请读者批评指正。

译者

2003 年 4 月

前 言

对于为用户开发应用程序的专业人士来说，掌握解题方法和编程概念是非常必要的。遗憾的是，对于许多读者来说，解题方法比计算机语言的语法更难于掌握。编程的艺术在于能掌握各种各样的技术，并能用这些技术去解决特定的问题。读者应该首先掌握基本的编程方法和解题方法，然后在学习特定编程语言的过程中着重学习语法。可以在介绍解题方法的课程或第一门针对解题的语言课程中学习这种技巧。它可以减少学生在学习过程中的阻力，提高成功率。

本书供编程专业学生第一学期使用，它是一门介绍性课程，可作为必修或辅助教材使用。虽然本书是为初学者或没有接触过计算机的读者编写的，但其介绍的内容对学习计算机语言的人也很有帮助。

本书通过详细的解释和大量的实例逐步展开介绍，从基本的数学函数和操作符到代码、数组、指针和其他数据结构等技术的设计和使用的，以及数据库和面向对象程序设计的概念。书中用到了很多解题工具，包括问题分析图（problem analysis chart）、交互图（interactivity chart）、IPO 图、算法和流程图，来设计问题的解决方案。另外，附录中还介绍了一些辅助工具，包括通用建模语言（Universal Modeling Languages, UML）概念、Nassi-Schneiderman 图和 Warnier-Orr 图。最后的综合实例部分使用前面介绍过的概念给出了一个完整解题方法。在某些实例中，我们会使用一些高级技术改进前面介绍过的方法。本书中所列举的实例在商业应用中都极具代表性，给读者提供了极好的经验。这些例子所涉及的问题可能会在以后语言课程学习中遇到，那时读者就可以通过上机将这些知识付诸实践。

本书的编排

第 I 部分“解题和编程方法简介”给出了解题的基本概念。介绍了计算机如何解决问题，以及用计算机解决问题时所需的数学概念、分析问题和解决问题的步骤。第 II 部分“逻辑结构”介绍了基本的编程概念，包括局部和全局变量、参数和 4 种基本的逻辑结构。这 4 种基本的逻辑结构是顺序结构、分支结构、循环结构和 case 逻辑结构。该部分还介绍了如何使用这些结构进行编程。第 III 部分“数据结构”介绍了数组的概念、排序技巧、查找方法、堆栈、链表、二叉树和数据库。第 IV 部分“面向对象编程”介绍了用面向对象程序设计语言解决问题的基本概念。第 V 部分“文件处理”对学习 COBOL 的读者来说是相当重要的。虽然数据库工具正在取代传统的处理海量数据的技术，然而政府和业界仍然有许多 COBOL 程序需要维护。第 VI 部分“应用软件中解题方法的使用”介绍了如何使用这些程序来撰写报告和进行预算，还介绍了如何综合利用这些程序来建立文档或描述问题。该部分还介绍了字处理、设计、电子数据表和信息处理的概念。附录介绍了商业中常用到的模板、排序和合并方法、ASCII 码和 EBCDIC 码、空白表格以及其他的解题工具和 UML 的概念。

致 谢

感谢那些审查原稿并提出了建议和建设性意见的人。特别感谢 Philip Davis (DelMar College, Texas), Richard Windley (ECPI College of Technology, Virginia) 和 Costas Vassiliadis (Ohio University)。

另外, 还要感谢 Mayda Bosco, Alex Wolf (Prentice Hall) 和 Megan Smith-Creed (Custom Editorial Productions, Inc) 在整个过程中所给与的帮助和支持。

还要感谢 Jeremy Stanton (College of the RedWoods 的学生) 为本书设计了封面。

目 录

第 I 部分 解题和编程方法简介

第 1 章 解决问题的一般概念	3
1.1 日常生活问题的解决	3
1.2 问题的类型	4
1.3 计算机解决的问题	5
1.4 解决问题的难点	5
1.5 本章小结	5
1.6 术语	6
1.7 简答	6
1.8 习题	6
1.9 纠错	10
第 2 章 计算机解决问题的初级概念	11
2.1 常量和变量	12
2.2 数据类型	14
2.3 函数	18
2.4 操作符	20
2.5 表达式和等式	23
2.6 本章小结	29
2.7 术语	29
2.8 简答	29
2.9 习题	31
2.10 纠错	33
第 3 章 编程概念	35
3.1 计算机如何存储数据	36
3.2 计算机通信	37
3.3 问题的求解	37
3.4 使用这些工具	48
3.5 测试解决方案	48
3.6 编程	48
3.7 本章小结	48
3.8 术语	49
3.9 简答	49
3.10 习题	49

第 II 部分 逻辑结构

第 4 章 程序结构	57
4.1 构建程序结构的要点	57
4.2 内聚和耦合	59
4.3 模块及其功能	60
4.4 局部变量和全局变量	61
4.5 参数	62
4.6 变量命名和数据字典	65
4.7 4 种逻辑结构	66
4.8 本章小结	67
4.9 术语	68
4.10 简答	68
4.11 习题	68
第 5 章 采用顺序逻辑结构解决问题	69
5.1 算法指令和流程图符号	69
5.2 顺序逻辑结构	71
5.3 解决方案开发	72
5.4 本章小结	78
5.5 简答	79
5.6 习题	79
5.7 纠错	80
第 6 章 采用分支结构解决问题	81
6.1 分支逻辑结构	81
6.2 多重 IF/THEN/ELSE 指令	83
6.3 使用串行逻辑	84
6.4 使用正逻辑	86
6.5 使用负逻辑	89
6.6 逻辑转换	91
6.7 选择哪一种分支逻辑	91
6.8 分支表	96
6.9 综合实例	100
6.10 本章小结	106
6.11 术语	108

6.12 简答	108	9.5 综合实例	156
6.13 习题	108	9.6 多维数组	159
6.14 纠错	110	9.7 查表技术	163
第7章 采用循环结构解决问题	111	9.8 指针技术	168
7.1 循环逻辑结构	111	9.9 综合实例	179
7.2 增量	112	9.10 本章小结	184
7.3 累加	112	9.11 术语	184
7.4 WHILE/WHILE-END	112	9.12 简答	187
7.5 综合实例	114	9.13 习题	187
7.6 REPEAT/UNTIL	115	9.14 纠错	188
7.7 综合实例	117	第10章 数据结构	189
7.8 自动计数循环	118	10.1 文件概念入门	189
7.9 综合实例	121	10.2 排序技术	190
7.10 嵌套循环	122	10.3 栈	197
7.11 指示器	124	10.4 以记录作为一种数据结构	197
7.12 算法指令和程序流程图符号	124	10.5 链表	199
7.13 递归	126	10.6 二叉树	211
7.14 本章小结	128	10.7 本章小结	225
7.15 术语	128	10.8 术语	225
7.16 简答	128	10.9 简答	225
7.17 习题	128	10.10 习题	225
7.18 纠错	129	第11章 关系数据库管理系统	229
第8章 采用 Case 结构解决问题	131	11.1 关系数据库管理系统	229
8.1 Case 逻辑结构	131	11.2 表格	230
8.2 代码	132	11.3 主键、辅键、外键	231
8.3 综合实例 I	133	11.4 范式化表格	231
8.4 综合实例 II	134	11.5 模式	236
8.5 本章小结	134	11.6 创建表格	236
8.6 术语	135	11.7 查询	238
8.7 简答	135	11.8 窗体设计	239
8.8 习题	135	11.9 界面设计	239
8.9 纠错	136	11.10 报表	239
		11.11 使用 RDBMS 设计解决方案	240
		11.12 关系数据库管理系统实例： 家庭财务管理	242
第III部分 数据结构		11.13 本章小结	247
第9章 处理数组	141	11.14 术语	247
9.1 数组	141	11.15 简答	247
9.2 一维数组	143	11.16 习题	248
9.3 综合实例	147		
9.4 二维数组	151		

第IV部分 面向对象编程

第12章 面向对象程序设计的概念..... 253

- 12.1 面向对象程序设计 253
- 12.2 图形用户界面 256
- 12.3 事件驱动的面向对象程序设计 257
- 12.4 本章小结 258
- 12.5 术语 258
- 12.6 简答 258
- 12.7 习题 258

第13章 面向对象程序设计 259

- 13.1 面向对象的应用程序设计 259
- 13.2 界面设计 264
- 13.3 事件驱动的面向对象程序设计 265
- 13.4 本章小结 270
- 13.5 术语 270
- 13.6 简答 270
- 13.7 习题 270

第V部分 文件处理

第14章 文件的概念 275

- 14.1 顺序访问文件 275
- 14.2 随机访问文件 276
- 14.3 缓冲器 276
- 14.4 主键和辅键 276
- 14.5 算法指令和流程图符号 276
- 14.6 系统流程图 278
- 14.7 记录设计 278
- 14.8 顺序访问文件的处理 282
- 14.9 初始读 282
- 14.10 随机访问文件的处理 282
- 14.11 本章小结 284
- 14.12 术语 284
- 14.13 简答 284
- 14.14 习题 284

第15章 顺序访问文件应用程序 287

- 15.1 设计输出报表 287
- 15.2 标题计数器 and 行计数器 288
- 15.3 控制中断 288

15.4 多重控制中断 295

15.5 使用指示器进行程序控制 296

15.6 错误处理 298

15.7 空文件 300

15.8 本章小结 300

15.9 术语 306

15.10 简答 307

15.11 习题 307

第16章 顺序访问文件的更新 317

16.1 创建文件 318

16.2 主文件 318

16.3 事务文件 318

16.4 活动文件 318

16.5 备份文件 318

16.6 使用事务文件更新主文件 318

16.7 综合实例 324

16.8 另一种有效的方法 335

16.9 本章小结 335

16.10 术语 335

16.11 简答 335

16.12 习题 335

第17章 随机访问文件的处理和更新 341

17.1 索引文件 341

17.2 使用交互式方法更新随机访问文件 341

17.3 本章小结 356

17.4 术语 356

17.5 简答 357

17.6 习题 357

第VI部分 应用软件中解题方法的使用

第18章 应用软件介绍 361

18.1 水平型软件的类型 361

18.2 学习新软件包的步骤 362

18.3 软件的兼容性问题 363

18.4 文件组织 363

18.5 本章小结 364

18.6 术语 364

18.7 简答 364

18.8 习题 364

第 19 章 字处理和桌面出版软件的使用	365	20.9 简答	394
19.1 字处理	365	20.10 习题	394
19.2 图形	368	第 21 章 文档处理	397
19.3 桌面出版软件	368	21.1 计算机系统的 6 大组成部分	397
19.4 一些设计思想	368	21.2 信息处理	398
19.5 字处理和桌面出版设计	371	21.3 软件套件	398
19.6 文档类型	373	21.4 共享数据	399
19.7 本章小结	376	21.5 对象链接和嵌入技术	399
19.8 术语	377	21.6 本章小结	402
19.9 简答	377	21.7 术语	402
19.10 习题	377	21.8 简答	402
第 20 章 使用电子数据表解决问题	379	21.9 习题	402
20.1 电子数据表的基础	379	附录 A 商业应用中的常用公式	407
20.2 电子数据表的设计	382	附录 B 排序、合并和查找方法	409
20.3 综合实例	388	附录 C 数据表示的 ASCII 代码和	
20.4 宏	388	EBCDIC 代码	415
20.5 图表与图表的制作	390	附录 D 解决问题时要使用的图	419
20.6 综合实例	392	附录 E 其他的问题解决工具	431
20.7 本章小结	393	附录 F 统一建模语言概述	435
20.8 术语	394	附录 G 术 语 表	441

第 I 部分 解题和编程方法简介

- 第 1 章 解决问题的一般概念
- 第 2 章 计算机解决问题的初级概念
- 第 3 章 编程概念

第 1 章 解决问题的一般概念

本章要点

日常生活问题的解决
问题的类型
计算机解决的问题
解决问题的难点

学习目标

通过本章的学习，您应该能够做到：

- (1) 了解启发式方案和规则算法方案的不同。
- (2) 列举和描述规则算法方案在解决问题时的 6 个步骤。
- (3) 用这 6 个步骤解决问题。

1.1 日常生活问题的解决

生活中的每一天，人们都会遇到许多需要做出选择的问题。它们可能是选择电视节目这种简单问题，也可能和选择新职业一样重要。如果做出了错误的选择，就会浪费时间和资源。所以知道如何做出正确的选择是相当重要的。遵循下面 6 个步骤可以确保做出最好的选择。步骤（问题解决方法里的 6 步）如下：

1. 明确问题

解决问题的第一步是明确问题。在课堂上，大部分问题都是明确的，以作业或问题的形式在课本上给出。然而，在课堂外，我们必须在开始解决问题之前首先明确问题。如果连问题是什么都不知道，又如何解决它呢？

2. 理解问题

接下来，必须先搞清楚问题所涉及的方面。其中包括理解问题提出方的知识背景。也就是说，当为用户制定一套解决方案的时候，我们必须了解用户的知识范围。根据不同知识范围确定使用不同的结构。例如，当一个对城市非常不熟的人询问如何去饭店时，我们就必须详细地告诉他每一个细节。当使用电脑解决问题时，电脑所能理解的有限语言和指令就是它的知识背景。超出这个范围的指令就不能使用，所以了解知识背景是很重要的。了解我们自己的知识背景同样重要。如果不了解某个学科，我们就不可能解决其中的问题。例如，要解决微积分的问题，就必须了解微积分；要解决会计学中的问题，就必须了解会计学。关键是我们必须能与客户沟通并搞清解决问题涉及的细节。

3. 寻找备选方案

尽可能全面地列出备选方案。可以征求不同人的意见以找到不同的解决方案，当然这些方案必须是可行的。例如，你从美国西海岸出发绕过太平洋、亚欧大陆和大西洋去东海岸是可以的，但这样做显然是不可行的。

4. 从备选方案列表中找到最好的解决方案

在这一步中，必须在选择前明确并评价每种方案的利弊。所以必须制定一个评定的标准。根据这些标准对所有的方案进行评价。

5. 列出所选择的解决方案的指令

这些有限的、分步的指令必须包含在第二步所确定的知识范围内。任何人或机器无法

理解的指令都不能使用。特别是在使用电脑进行工作时，这种限制就更加严格了。

6. 评价解决方案

评价或测试解决方案就是检查它的结果是否正确，是否令用户满意（如果一个人需要一个用来睡觉的家具，我们可以给他买一个轻便的小床，但这不一定能让他非常满意。）如果结果错误或者不令人满意，问题解决人员必须重新设计一个解决方案。

如果 6 步中的任何一步没有圆满地完成，那么结果往往会比期望值低。

人们每天都要解决家庭中的、工作中的或者是其他任何所到之处存在的问题。家庭中的问题可能是正餐吃什么、今晚看什么电影、买哪一辆车或是怎么卖房子等等。工作中的问题可能是搞好同事关系、制定工作策略、搞好管理或是处理好与客户的关系等等。一个员工做出的选择越好，他对公司就越有价值。解决问题的这 6 个步骤可以适用于任何情况。大多数人可能不知道这 6 个步骤，但却一直在使用它们。

下面这个例子是关于今晚做什么的问题。

1. 明确问题

如何度过一个漫长的夜晚。

2. 理解问题

虽然只是个简单的问题，我们还是要弄清大家的知识背景。只能选择那些大家都会的娱乐方式。如果有人不会下国际象棋，就不能选择这种方式。

3. 寻找备选方案

- (1) 看电视
- (2) 邀请朋友
- (3) 玩游戏
- (4) 玩迷你高尔夫
- (5) 去娱乐场所
- (6) 参加朋友的聚会

当你再也想不出其他方案的时候，这个列表就算完成了。

4. 从备选方案列表中找出最好的解决方案

- (1) 去掉不可行的方案，比如那些花钱太多或者是有人不感兴趣的方案。
- (2) 说明余下方案的利弊。
- (3) 权衡利弊并获得最终方案。如果所有的步骤都圆满完成，那么将得到最好的方案。

5. 准备一个步骤（指令）列表，它会使我们度过一个美好的夜晚

6. 评价解决方案

我们过得快乐吗？

通过这些步骤，问题解决者可以确信是否做出了最好的选择，是否达到了预期的效果。

1.2 问题的类型

问题并不总有简单的解决方法。一些问题，比如平衡收支或烤蛋糕等，可以通过一系列的动作来解决。这些解决方案叫做**算法方案**。一旦其他备选方案被排除，例如，一旦找到了一个最好的平衡收支方法，就可以通过选定的步骤达到预期的目标。这些步骤叫做**算法**。还有一些问题，例如怎样买绩优股或是否扩大公司规模，也不能直截了当地解决。要解决这些问题除了必须有相应的知识和经验外，还需要经过不断的尝试和失败才能达到最终目标。这种不能通过一些直观的步骤来解决的方案叫做**启发式方案**。

问题解决者在算法方案和启发式方案中都可以采取上述 6 个步骤。然而，在启发式方案中，第 6 步的正确性和恰当性标准相对不确定。我们很容易评价一个平衡收支问题的解决方案是否正确并令人满意，但很难确定是否买到了绩优股。在启发式方案中，问题解决者通常要反复执行这 6 个步骤，并在做出决定前仔细评价每一种可能的方案。另外，同一种方案不一定在任何时候都适用，所以问题解决者在以后解决同样问题的时候必须重新进行评价和分析。总之，要实事求是，具体问题具体分析。

许多问题需要将这两种方案结合起来加以解决。

1.3 计算机解决的问题

在本书中，**解决方案**这个术语是指问题解决过程中的第 5 步所列出的指令序列，遵从这些指令才能得到最满意的结果。**结果**是指成果或在计算机辅助下得到的完整答案。**程序**是指用特定的计算机语言编写的一组用于解决问题的指令。

计算机主要用来执行那些对于人来说非常困难或非常耗时的算法方案。而人类比计算机更善于使用启发式方案。解决一个复杂的微积分问题或者将 10 000 个名字按字母排序，用计算机来处理就显得非常简单，但是要让计算机明白怎样丢出一个球或者怎样说英语就很困难了。难点在于程序的编写。怎样才能用一些计算机可以理解的步骤来描述扔球和说英语的过程呢？

处理这类启发式问题所涉及的计算机技术领域叫做人工智能。人工智能可以让计算机建立自己的知识库并学会人类的语言。最终，计算机解决问题的能力将会和人相差无几。人工智能是处于发展中的计算机技术领域。

在计算机可以像人类一样思考以前，仍是由人类处理大部分启发式的问题，而计算机主要处理一些算法的问题。因此，本书将只涉及算法方案。启发式方案有助于方案的选择，但它必须首先转化成算法形式才能供计算机使用。

1.4 解决问题的难点

人们在解决问题时会遇到很多困难。一些人没有学习如何解决问题，一些人又害怕作出错误的选择。因此在解决问题的过程中，人们通常不能很好地完成其中的一步或几步。他们可能错误地定义了问题或者没有列出足够的选择方案。在选择最优方案时，他们又可能排除了好的方案，或者草率地估计了利弊。他们还可能搞错了整个步骤的顺序，或者在构建问题的整体框架前过早地专注细节。最后，他们还可能错误或草率地评价解决方案。

问题解决的过程不容易掌握，需要花大量的时间和实践进行完善，但长期积累的经验证明，这种过程在解决问题时是非常有效的。

用电脑解决问题时，最难的事情之一就是编写指令。例如，从 3 个数中找出最大的一个。几乎所有的人都能立刻说出哪个数最大，但许多人都说不清是怎样做出这种判断的。许多人会说：“我不能解释我是如何知道的，但我就是知道。”这种解释对计算机来说没有什么用处。计算机是一种工具，它只能执行用户解释清楚的任务。

计算机有它自己特殊的通信系统，无论是程序员还是用户都必须了解它。这套系统要求我们详细阐明解决方案中的每一个步骤而且排序正确。要知道，除非我们告诉它，否则计算机一无所知，然而虽然它很无知，但它对解决问题却很有帮助。

1.5 本章小结

问题解决过程中的 6 个步骤能引导我们找到最好的解决方案。它们包括：(1) 明确问题，(2) 理解问题，(3) 寻找备选方案，(4) 从备选方案列表中找出最好的解决方案，(5) 列出所选择的解决方案的指令，(6) 评价解决方案。问题的解决方案分为启发式方案和算方案。算法方案通过一系列的步骤来实现。启发式方案通过不断的实践和失败来实现。算

法方案比启发式方案更容易被计算机理解。

问题解决过程中有可能出现一些错误。有时问题解决者没有正确定义问题或者没有完成某些步骤。有些时候，备选方案列举得不够充分或给出的计算机指令有错误。有时候过早地专注于细节而忽略了整体框架，或者没有对方案进行测试。如果问题解决的过程没有圆满地完成，就不会得到预期的结果。

这些问题解决的步骤既可以用于日常生活中，也可以用在计算机上。好的问题解决技巧可以让你逻辑客观地看待问题，从而节约时间和其他资源。

1.6 术 语

算法 (algorithm)

算法方案 (algorithmic solution)

启发式方案 (heuristic solution)

程序 (program)

结果 (Results)

解决方案 (solution)

1.7 简 答

1. 解决问题的 6 个步骤的是什么？
2. 什么是问题的算法方案？
3. 列举 3 个目前生活中遇到的可以用算法方案解决的问题。解释为什么每个问题在本质上都是属于算法的。
4. 什么是问题的启发式方案？
5. 列举 3 个目前生活中遇到的可以用启发式方案解决的问题。解释每个问题有什么启发性。
6. 列举 3 个在家里、学校或商业中可能遇到的，使用计算机辅助解决比较有效的问题。这些问题需要算法方案或启发式方案吗？为什么？

1.8 习 题

1. 选择在上面第 3 题中列举的一个问题，将问题解决的 6 个步骤补充完整。格式如下：
 - 第 1 步：明确问题。
 - 第 2 步：理解问题。
 - (1) 在有助于问题理解的地方进行注释。
 - (2) 描述知识背景（这一项包括解决问题所需要的各种知识）。
 - 第 3 步：寻找备选方案。

方案	优点	缺点
(1)		
(2)		
(3)		
.....		
 - 第 4 步：从备选方案列表中找到最好的解决方案。
为什么选择这个方案？
 - 第 5 步：列出所选择的解决方案的指令。
 - (1)
 - (2)
 - (3)
 -
 - 第 6 步：测试这个方案。
这个方案有效吗？

如果无效, 如何进行改进?

2. 针对下列每一项任务, 写出一个有编号的、详细步骤的指令(方案), 要尽量详细, 使其他人不问任何问题就可以独立完成工作。给出所需要的知识背景。例如, 给出一个准备一杯可可饮料的任务, 其知识背景可能包括对如下物品的了解: 牛奶或水、冰箱、盘子、勺子、可可粉、茶杯、壁炉或微波炉等等。
 - (1) 准备一杯可可饮料
 - (2) 削铅笔
 - (3) 从教室走到学生休闲室、宿舍或自助餐厅
 - (4) 发动一辆车(包括车发动不了时该怎么办)
 - (5) 从厨房里取一杯水
3. 详述下列问题的知识背景。
 - (1) 收支平衡
 - (2) 开车
 - (3) 修车
 - (4) 建房
 - (5) 计算一个月的养车费用
4. 写出从3个数中找出最大数的方案。详细列出每一步, 使其他人可以使用该方案找到最大的那个数。
5. 这是个关于机器人 Otto 的问题。Otto 不是非常聪明。实际上, 它的知识背景只有15条指令。然而, 它可以根据设定的指令活动, 只要这些指令在它知识背景的15条指令范围内。它按顺序执行指令, 也就是执行完一条指令后, 紧接着执行下一条指令。它只按设定好的指令行事。下面列出了 Otto 的指令集、假定、一个示例解决方案和问题。

Otto 的知识背景包括如下指令:

身体指令

- (1) Stand up (起立)
- (2) Sit down (坐下)
- (3) Take a step (向前走一步——必须在站立时执行)
- (4) Turn (向右转 90 度——必须在站立时执行)
- (5) Raise arms (举起手臂, 向前抬到与身体成直角)
- (6) Lower arms (将手臂放到体侧)

算术指令

Otto 的内存中有一个数, 它可以将这个数加一或减一。指令开始执行时这个数是零。由于存储区域中的值可以改变, 所以可以把这个数看成一个变量, 常量是没有存储地址的。

- (7) Add one (将内存中的数加一)
- (8) Subtract one (将内存中的数减一)
- (9) Goto s (跳到 s 号指令, 其中 s 是一个已知数, 然后从 s 开始继续执行。
s 可能在当前位置以前或以后)

例如:

20. Goto 23

它的意思是跳到 23 号指令继续执行。