



国外经典教材·计算机科学与技术



Java Software Structures: Designing and Using Data Structures

数据结构 (Java版)

(美) John Lewis 著
Joseph Chase 译
施平安



清华大学出版社

国外经典教材·计算机科学与技术

数 据 结 构 (Java 版)

John Lewis
(美) Joseph Chase 著

施平安 译



清华大学出版社

北 京

内 容 简 介

本书是数据结构课程的优秀教材。主要讲述如何在正确的软件工程原则指导下,用精心定义的数据结构和算法实现高质量的程序。为使读者能更好地理解本书,各章在内容安排上从易到难,对于每章所涉及的数据结构,都先介绍其概念,然后举例说明其用途,再讨论如何用 Java 编程语言实现,最后对各种实现的效率进行分析、对比。

Simplified Chinese edition copyright © 2004 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

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

Original English language title: Java Software Structures: Designing and Using Data Structures by John Lewis, Joseph Chase, Copyright © 2004

EISBN: 0-201-78878-0

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: 01-2003-5407

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

数据结构 (Java 版) / (美) 李维斯 (Lewis, J.), (美) 查兹 (Chase, J.) 著; 施平安译. —北京: 清华大学出版社, 2004.

(国外经典教材·计算机科学与技术)

书名原文: Java Software Structures: Designing and Using Data Structures

ISBN 7-302-08552-8

I. 数… II. ①李… ②查… ③施… III. ①数据结构—教材 ②JAVA 语言—程序设计—教材

IV. ①TP311.12 ②TP312

中国版本图书馆 CIP 数据核字 (2004) 第 037824 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: 徐刚

封面设计: 久久度文化

印 刷 者: 北京市清华园胶印厂

装 订 者: 三河市金元装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 30 字数: 711 千字

版 次: 2004 年 6 月第 1 版 2004 年 6 月第 1 次印刷

书 号: ISBN 7-302-08552-8/TP·6136

印 数: 1~4500

定 价: 49.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770175-3103 或 (010)62795704。

出版说明

近年来,我国的高等教育特别是计算机学科教育,进行了一系列大的调整 and 改革,急需一批门类齐全、具有国际先进水平的计算机经典教材,以适应当前我国计算机科学的教学需要。通过使用国外先进的经典教材,可以了解并吸收国际先进的教学思想和教学方法,使我国的计算机科学教育能够跟上国际计算机教育发展的步伐,从而培育出更多具有国际水准的计算机专业人才,增强我国计算机产业的核心竞争力。为此,我们从国外知名的出版集团 Pearson 引进这套“国外经典教材·计算机科学与技术”教材。

作为全球最大的图书出版机构, Pearson 在高等教育领域有着不凡的表现,其下属的 Prentice Hall 和 Addison Wesley 出版社是全球计算机高等教育的龙头出版机构。清华大学出版社与 Pearson 出版集团长期保持着紧密友好的合作关系,这次引进的“国外经典教材·计算机科学与技术”教材大部分出自 Prentice Hall 和 Addison Wesley 两家出版社。为了组织该套教材的出版,我们在国内聘请了一批知名的专家和教授,成立了一个专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动,各位委员根据国内外高等院校计算机科学及相关专业的现有课程体系,并结合各个专业的培养方向,从 Pearson 出版的计算机系列教材中精心挑选针对性强的题材,以保证该套教材的优秀性和领先性,避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量,我们为这套教材配备了一批经验丰富的编辑、排版、校对人员,制定了更加严格的出版流程。本套教材的译者,全部来自于对应专业的高校教师或拥有相关经验的 IT 专家。每本教材的责编在翻译伊始,就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华,在经过翻译、排版和传统的三审三校之后,我们还请编审委员或相关的专家教授对文稿进行审读,以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限,该套教材在出版过程中很可能还存在一些遗憾,欢迎广大师生来电来信批评指正。同时,也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材,共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

2004.03.20

国外经典教材·计算机科学与技术

编 审 委 员 会

主任委员:

孙家广 清华大学教授

副主任委员:

周立柱 清华大学教授

委员 (按姓氏笔画排序):

王成山	天津大学教授
王 珊	中国人民大学教授
冯少荣	厦门大学教授
冯全源	西南交通大学教授
刘乐善	华中科技大学教授
刘腾红	中南财经政法大学教授
吉根林	南京师范大学教授
孙吉贵	吉林大学教授
阮秋琦	北京交通大学教授
何 晨	上海交通大学教授
吴百锋	复旦大学教授
李 彤	云南大学教授
沈钧毅	西安交通大学教授
邵志清	华东理工大学教授
陈 纯	浙江大学教授
陈 钟	北京大学教授
陈道蓄	南京大学教授
周伯生	北京航空航天大学教授
孟祥旭	山东大学教授
姚淑珍	北京航空航天大学教授
徐佩霞	中国科学技术大学教授
徐晓飞	哈尔滨工业大学教授
秦小麟	南京航空航天大学教授
钱培德	苏州大学教授
曹元大	北京理工大学教授
龚声蓉	苏州大学教授
谢希仁	中国人民解放军理工大学教授

译 者 序

随着 Java 语言的普及,已经有越来越多的程序开发人员喜欢上这种简单易懂的面向对象语言。与此同时,随着学习和应用的深入,数据结构和算法设计的重要性愈发显现出来。对于以前接触过 C 或 C++ 的程序员而言,使用 C 或 C++ 开发诸如队列、堆栈、链表和树等数据结构并非难事。遗憾的是,现在介绍 Java 数据结构和算法的好书却较少,使大家在进一步学习、使用 Java 的过程中遇到了一些困难。本书正是为此目的而撰写的。作者使用浅显易懂的语句来阐述一个个常见的数据结构,并配以大量的例子和代码来帮助大家理解如何使用 Java 实现这些数据结构。此外,本书还从软件工程的高度来分析、比较各种数据结构的实现效率,指出各种算法的优缺点。特别值得一提的是,每章都提供了大量的习题供读者复习时使用。附录 A 简要地阐述了 Java 面向对象概念,以帮助大家能更好地学习本书的内容;而附录 B 提供了 Java 类库中许多常用类的参考资料,使大家能够更容易地开发出自己的 Java 程序。

这确是一本介绍 Java 数据结构的好书。为使读者能更好地阅读本书,各章在内容的安排上不仅从易到难,而且在组织形式上力求保持一致:对于每章所涉及的数据结构,都先介绍它们的概念,接着举例说明它们的用途,然后讨论如何实现它们,最后分析比较各种实现的效率。为方便读者的学习,书中所有例子的源代码都可下载(www.aw.com/cssupport)。

全书的翻译出版是集体工作的结晶。柳赐佳、周莎莎、施晓东、施惠琼、蔡桂凌、施琳琼、陈华、柳晔锦、柳晔惠、施卓成、张琼雯、张庭辉、方杰等负责全书的翻译工作,柳小艳、孔颂燕、梁锦伦等负责全书的审校工作,施金庭、柳聿荫、施群肖和缪彩珠等负责全书的录入和排版工作。全书最后由施平安负责统稿。

在翻译过程中,我们对书中出现的所有术语和难词难句都进行了仔细的推敲和研究,然而有些问题在译者本人的研究领域中也不曾遇到过,疏漏和争议之处在所难免,望广大读者提出宝贵的意见。

译者

2004 年 2 月 10 日

前 言

本书的目标是成为数据结构与算法课程的教材。因为该课程一般作为计算课程体系中的第二门课，所以通常把它称为 CS2 课程。我们在设计本教材时力求体现计算技术课程的宗旨。

从教学方法上看，本教材承袭了 John Lewis 和 William Loftus 编著的前导课 CS1《Java 程序设计基础》（本书已由清华大学出版社出版）的风格和方法，本教材借鉴了其强调的许多特征。这两本书都以严格而一致的方法倡导了正确的学习顺序。

这就是说，本书假定学生没学过《Java 程序设计基础》。因此，在课程中可能涉及的材料（诸如递归或者排序），都在本书中进行了介绍。附录 A 概述了面向对象概念以及如何用 Java 实现它们。该附录可以作为复习材料，或使具有不同基础的学生以最快的速度熟悉它们。

我们了解 CS2 课程在专业课中的关键作用，并认为本教材能够很好地满足该课程的需要。

教学方法

这类教材在总体教学方法上差异很大。教学方法建立在我们极力强调的一个重要原则之上。首先，介绍了一贯需要探讨的各种集合。其次，强调了健壮的软件设计技术的重要性。最后对本书进行了组织以支持和巩固重要构想：数据结构和算法的研究。让我们进一步研究这些原则。

一致性

在讨论某类集合时，我们依次解决如下问题：

- 概念：从概念上讨论集合，建立它所提供的服务，即接口
- 用法：用例子说明如何使用集合的特定性质解决问题，而不管如何实现
- 实现：探讨了集合的各种实现的选择
- 分析：对集合的实现进行了比较

在讨论中根据需要插入 Java 集合 API。如果支持 API 中的某种特定集合类型，则对其和实现进行讨论。因此，我们讨论了 API，但并不力求彻底地讨论。而且我们会毫不犹豫地指出它的不足。

高层次分析。我们在第 1 章中建立了大 O 表示法的概念，并在全书中一直使用它。这种分析法比数学分析法更直观。

健壮的程序设计

纵观全书,我们始终优先考虑健壮的软件工程实践。集合实现的设计及使用它们的程序的设计,这些都遵循一致而正确的标准。

最重要的是集合的接口与底层实现相分离。一个其功能提供的服务总是在 Java 接口中正式定义的。每当需要以抽象形式加强其功能时,就把接口名用作集合的类型名。

除了遵循严格的设计原则外,我们在全文的讨论中一直强调它们。我们力求通过例子和反复的强调来讲授设计原则。

清楚的结构

本书内容经过仔细编排,使离题的可能性最小,这样还可以强化本书的写作意图。这种编排使本书不仅可以作为宝贵的参考资料,而且还可以在数据结构与算法的教学研究中起作用。

本书可以分成三部分:第一部分由前 5 章组成,讨论了影响数据结构与算法的各种问题。第二部分包括第 6~8 章,讨论了线性集合(堆栈、队列和链表)。第三部分包括第 9~14 章,讨论了非线性集合(树、堆栈、散列和图)。除了树以外,其他各种集合类型均以单独一章进行讨论。树分散在若干章节中讨论,探讨了树在各个方面的应用问题。

附录 A 讨论了基本的面向对象概念,以及如何用 Java 实现这些概念。对于以前曾学过此内容的学生,它可以作为复习资料,而且既可以在讨论本书的主要内容之前介绍,也可以根据需要在每个主题之前介绍。对于那些在其他语言(诸如 C++ 或者 C#)中学过面向对象概念,并且需要了解如何用 Java 实现那些概念的学生,本附录也可以作为专用的 Java 教程。

附录 B 是 Java 类库中许多常用类的参考资料。学生手头拥有此信息,更容易开发他们的程序。

各章节剖析

第 1 章讨论了软件质量的各个方面,并概述了软件开发问题。本章的设计目标是,在具体讨论数据结构与算法设计之前,培养正确的思维习惯。本章还介绍了算法分析基础的基本概念。

第 2 章建立了集合的概念,强调了接口与实现分离的必要性。作为一个例子,介绍了袋集合,并讨论了一种基于数组的实现。

第 3 章讨论了使用引用建立链式数据结构。讨论了与链表管理有关的基本问题,然后使用基本的链式数据结构,重新定义了(第 2 章中介绍的)袋集合的另一种实现。

第 4 章简要介绍了递归的概念,以及怎样才能优雅地进行递归求解。本章研究了递归的具体实现细节,并讨论了分析递归算法的基本思想。

第 5 章讨论了线性查找算法和折半查找算法,同时还讨论了几种排序算法:选择排序、插入排序、冒泡排序、快速排序和归并排序。本章强调了与查找和排序有关的编程问题,诸如使用 `comparable` 接口作为对象比较的基础。第 5 章后面讨论了基于特定数据结构的查找和排序(诸如堆排序)。

第 6 章开始研究特定集合。我们从堆栈开始,从概念和实现上研究了各种集合。堆栈是一种非常直观的集合。本章同时讨论了基于数组的实现和基于链表的实现方法,然后对它们进行了比较。

第 7 章讨论了先进先出队列的概念和实现。作为有效使用队列的例子,讨论了基数排序。本章提供的实现包括基本的链表实现,以及固定循环数组实现。

第 8 章讨论了 3 种链表类型:有序链表、无序链表和索引链表。对这 3 种链表类型进行了比较,讨论了它们共有的操作和每种链表特有的操作。在各种链表的设计中正确地使用了继承,这些链表同时使用数组和链接表实现。

第 9 章概述了树,引入了关键术语和概念。本章讨论了树的各种实现方法,以及使用二叉树来表示和计算算术表达式。

第 10 章根据第 9 章中建立的基本概念来定义经典的二叉查找树。分析了二叉查找树的链接实现,然后讨论了树结点的平衡是性能问题的关键。由此引出了二叉查找树的 AVL 和红黑树实现的讨论。

第 11 章讨论了堆的概念、使用 and 实现。作为堆的实用性的例子,讨论了堆排序。本章同时讨论了堆的基于链接的实现和基于数组的实现。

第 12 章是前几章讨论的自然延伸。本章分析了 2-3 树、2-4 树和基本 B-树的概念,讨论了各种可选的实现方案。

第 13 章讨论了散列的概念及其相关问题。本章讨论了散列的各种 Java 集合 API。

第 14 章讨论了无向图和有向图的概念,并引入了重要的术语。本章讨论了几个常用的图算法,并讨论了各种实现方法,包括邻接矩阵。

对于需要复习基本的面向对象概念和了解如何用 Java 实现它们的读者,附录 A 的参考资料堪称首选。该附录包含了抽象、类、封装、继承和多态性的概念,以及许多相关的 Java 语言结构,诸如接口。

附录 B 是标准的 Java API 类库中常用类的参考资料。

辅助资料

本书的所有读者可以在 www.aw.com/cssupport 上得到如下辅助资料:本书所有程序的源代码;有资格的教师可以得到教学辅助资料:教师手册(包含作者的注解、教学提示等);本书部分练习题和编程题的答案;TestGen 测试库(包含试题中的问题);Powerpoint 幻灯片(介绍了本书的内容和附录 A);动画软件(演示了某些集合的处理过程)。有关如何得到它们的信息,请联系当地的 Addison-Wesley 销售代理,或者发电子邮件给 aw.cse@aw.com。

目 录

第 1 章 软件工程	1
1.1 软件开发	1
1.2 软件质量	2
1.3 软件开发生命期模型	6
1.4 统一建模语言	9
1.5 错误的处理	12
1.6 算法的分析	13
1.7 软件工程和数据结构	17
第 2 章 集合.....	21
2.1 本章简介.....	21
2.2 袋集合	24
2.3 使用袋的例子: bingo	28
2.4 袋的数组实现	30
2.5 分析袋的数组实现	42
第 3 章 链式结构	47
3.1 链式引用	47
3.2 管理链表	49
3.3 无链接的元素	51
3.4 袋的链式实现	52
3.5 分析袋的链式实现	59
第 4 章 递归	63
4.1 递归思想	63
4.2 递归编程	65
4.3 使用递归	68
4.4 分析递归算法	75
第 5 章 查找与排序	79
5.1 查找	79
5.2 排序	83
第 6 章 堆栈	95
6.1 堆栈 ADT	95
6.2 使用堆栈: 计算后缀表达式	97
6.3 使用堆栈: 模拟递归	103
6.4 堆栈的链式实现	104
6.5 堆栈的数组实现	107
6.6 实现堆栈: java.util.Stack 类	109
6.7 分析堆栈的实现	110
第 7 章 队列	115
7.1 队列 ADT	115

7.2 使用队列: 代码键	117
7.3 使用队列: 模拟售票器	120
7.4 使用队列: 基数排序法	124
7.5 队列的链式实现	127
7.6 队列的数组实现	130
7.7 使用循环数组来实现队列	132
7.8 队列实现的分析	135
第 8 章 链表	141
8.1 链表 ADT	141
8.2 使用有序链表: 制订赛程	147
8.3 使用索引链表: Josephus 问题	151
8.4 链表的数组实现	154
8.5 链表的链式实现:	158
8.6 分析链表的实现	162
第 9 章 树	169
9.1 树	169
9.2 树的实现策略	171
9.3 树的遍历	173
9.4 实现二叉树	176
9.5 使用二叉树: 表达式树	182
第 10 章 二叉查找树	193
10.1 二叉查找树	193
10.2 二叉查找树的链式实现	194
10.3 使用二叉查找树实现有序链表	204
10.4 平衡二叉查找树	205
10.5 实现二叉查找树: AVL 树	208
10.6 实现二叉查找树: 红黑树	210
10.7 实现二叉查找树: Java 集合 API	215
第 11 章 堆	221
11.1 堆	221
11.2 使用堆: 堆排序	225
11.3 使用堆: 优先队列	225
11.4 堆的链式实现	228
11.5 堆的数组实现	232
11.6 分析堆的实现	235
第 12 章 多叉查找树	239
12.1 合并树的概念	239
12.2 2-3 树	239
12.3 2-4 树	245
14.4 B-树	246
12.5 B-树的实现策略	248
第 13 章 散列	251
13.1 散列	251
13.2 散列函数	253
13.3 解决冲突	255
13.4 删除哈希表中的元素	259

13.5 Java 集合 API 中的哈希表	261
第 14 章 图	271
14.1 无向图	271
14.2 有向图	272
14.3 网络	274
14.4 图的通用算法	275
14.5 图的实现策略	281
附录 A Java 面向对象概念	285
A.1 面向对象简介	285
A.2 使用对象	286
A.3 类库和包	288
A.4 对象的状态和行为	289
A.5 类	290
A.6 封装	293
A.7 构造函数	295
A.8 方法重载	295
A.9 再谈引用	296
A.10 static 修饰符	301
A.13 继承	305
A.14 类层次体系	308
A.15 多态	311
A.16 异常	315
附录 B Java 类库	319
AbstractButton (javax.swing)	319
ActionEvent (java.awt.event)	320
AdjustmentEvent (java.awt.event)	321
Applet (java.applet)	322
ArrayList (java.util)	324
AWTEvent (java.awt)	325
BigDecimal (java.math)	326
BigInteger (java.math)	329
BitSet (java.util)	331
Boolean (java.lang)	332
BorderFactory (javax.swing)	333
BorderLayout (java.awt)	334
Box (javax.swing)	335
BoxLayout (javax.swing)	336
BufferedReader (java.io)	337
BufferedWriter (java.io)	338
ButtonGroup (javax.swing)	339
Byte (java.lang)	339
Calendar (java.util)	340
CardLayout (java.awt)	344
Character (java.lang)	345
Class (java.lang)	347
Color (java.awt)	349
Component (java.awt)	351
ComponentAdapter (java.awt.event)	355
ComponentEvent (java.awt.event)	356
Container (java.awt)	357
ContainerAdapter (java.awt.event)	359
ContainerEvent (java.awt.event)	359

Cursor (java.awt)	360
Date (java.util)	361
DateFormat (java.text)	362
DateFormatSymbols (java.text)	364
DecimalFormat (java.text)	365
DecimalFormatSymbols (java.text)	367
Dimension (java.awt)	368
Double (java.lang)	369
Error (java.lang)	370
Event (java.awt)	370
Exception (java.lang)	374
Float (java.lang)	374
FlowLayout (java.awt)	376
FocusAdapter (java.awt.event)	377
FocusEvent (java.awt.event)	377
Font (java.awt)	378
FontMetrics (java.awt)	379
Format (java.text)	380
Graphics (java.awt)	381
GregorianCalendar (java.util)	384
GridBagConstraints (java.awt)	385
GridBagLayout (java.awt)	387
GridLayout (java.awt)	389
Hashtable (java.util)	390
Image (java.awt)	391
ImageIcon (javax.swing)	392
InputEvent (java.awt.event)	393
InputStream (java.io)	394
InputStreamReader (java.io)	394
Insets (java.awt)	395
Integer (java.lang)	396
ItemEvent (java.awt.event)	397
JApplet (javax.swing)	398
JButton (javax.swing)	399
JCheckBox (javax.swing)	399
JCheckBoxMenuItem (javax.swing)	400
JColorChooser (javax.swing)	400
JComboBox (javax.swing)	401
JComponent (javax.swing)	402
JFileChooser (javax.swing)	403
JFrame (javax.swing)	404
JLabel (javax.swing)	405
JList (javax.swing)	406
JOptionPane (javax.swing)	408
JPanel (javax.swing)	410
PasswordField (javax.swing)	410
JRadioButton (javax.swing)	411
JScrollPane (javax.swing)	411
JSlider (javax.swing)	412
JTabbedPane (javax.swing)	413
JTextArea (javax.swing)	414
TextField (javax.swing)	415
JToggleButton (javax.swing)	416
JToolTip (javax.swing)	416
KeyAdapter (java.awt.event)	416
KeyEvent (java.awt.event)	417
Locale (java.util)	421
Long (java.lang)	423

Math (java.lang)	424
MessageFormat (java.text)	426
MouseAdapter (java.awt.event)	427
MouseEvent (java.awt.event)	428
MouseMotionAdapter (java.awt.event)	429
Number (java.lang)	429
NumberFormat (java.text)	430
Object (java.lang)	432
ParsePosition (java.text)	433
Point (java.awt)	433
Polygon (java.awt)	434
PrintStream (java.io)	435
Random (java.util)	436
Rectangle (java.awt)	437
Short (java.lang)	439
SimpleDateFormat (java.text)	440
SimpleTimeZone (java.util)	441
Stack (java.util)	442
String (java.lang)	442
StringBuffer (java.lang)	445
StringTokenizer (java.util)	447
System (java.lang)	448
SystemColor (java.awt)	449
Thread (java.lang)	452
Throwable (java.lang)	454
Timer (javax.swing)	455
TimeZone (java.util)	455
URL (java.net)	456
Vector (java.util)	457
Void (java.lang)	459
Window (java.awt)	460
WindowAdapter (java.awt.event)	461
WindowEvent (java.awt.event)	461

第1章 软件工程

本章目标

- 讨论软件开发的目的
- 确定软件质量的各个方面
- 分析几个软件开发生命期模型
- 探讨统一建模语言的表示法
- 分析与错误处理有关的问题
- 介绍算法分析的概念

软件开发的探讨必须从讨论与软件工程有关的问题开始。我们不应只是为了写程序而写程序，而是应该努力设计我们的软件。我们需要开发高质量的软件系统，能经得住用户和时间的考验。软件工程的原则将会带领我们实现该目标。本章讨论与软件工程有关的各种问题，并提出一些在讨论数据结构和软件设计时将会用到的关键术语。

1.1 软件开发

设想过河上一座桥的情形，这座桥是刚刚架设的。走近时看到一块标识牌，告之这座桥是由当地的建筑工人设计和建造的，并且没有桥梁工程师参与。你还会继续过这座桥吗？反之，如果告诉你桥梁是由工程师设计，并由建筑工人建造，会有什么差别吗？

在这个语境中，“工程师”一词指受过工程规范的历史、理论、方法和实践教育的个人。该定义包括电气工程、机械工程和化学工程等研究领域。软件工程是对高质量软件开发的基础技术和理论的研究。

“软件工程”这个术语早在 20 世纪 70 年代创造出来了，这是大势所趋——业界的领军人物在意识到已开发的大多数软件都质量不高的情况下提出的一个目标。他们希望开发人员改变过分简单化的程序编写思想，并朝着规范的工程思想发展。对于工程软件，我们首先必须意识到，该术语绝非仅仅是一种称号——实际上，它表示一种完全不同的态度。

关于软件工程是否已达到真正的工程规范的状态这一问题，已经引起了许多争论。我们将把那些争论留给软件工程课程吧。对我们来说，理解软件开发人员有着共同的历史、受到共同的理论约束就足够了。此外，为了协同工作，我们还必须了解当前的方法和实践。

归根结底，我们应该满足客户以及系统的最终用户的需要。客户指购买软件的人或者

组织。客户和用户在某些情况下可能是同一个人。

软件工程的目标与其他工程学科的目标极其相同：

- 解决正确的问题
- 在规定的时间和预算内交付一个解决问题的方案
- 交付一个高质量的解决方案

我们需要考虑错误问题，也许听起来比较奇怪，但是几乎每个项目都会在这方面出麻烦。经常会出现这种现象，软件开发人员将要交付产品时，不料竟会发现它并不能完全满足客户的需求。因此，任何软件开发过程的第一步是，确保掌握了待解决问题的详细信息。

关键概念：软件开发的第一步是分析问题，并开发出完整而精确的需求集。

问题分析牵涉到如下活动，诸如与客户面谈、观察现有过程和分析现有解决方案。根据这些活动开发的需求，不仅要确定解决方案必须提供的功能，诸如允许用户通过用户名和密码登录到一个系统上，而且还要确定如何支配这些功能及如何开发它们的约束条件，诸如可以用什么字符构成有效密码的说明。通过理解问题，我们能够更好地开发出一个解决正确问题的解决方案。

1.1.1 职业道德准则

作为专业人员，如果我们答应客户的请求，在某个时间以特定价格交付系统，那么我们必须在规定时间内交付软件。显然，如果企业不断地让客户失望，那么它肯定不能长期生存下去。但是这个问题超出了实际层面。工程规范的内容之一就是要能够制定精确的计划、进度和预算。不能在规定时间内交付软件，不仅会严重损害项目牵涉到的公司，而且还可能会影响我们的职业信誉。

关键概念：无论从实用性方面还是从理性方面考虑，软件工程师都必须在规定时间内和预算内交付软件。

真正的软件工程师恪守职业道德准则，这里涉及能力的概念。如果我们认为在那些需求下不能完成项目，那么我们有责任在确定需求时表达出来。

为了追求卓越的软件质量，我们必须认识到，质量对于不同的人有着不同的含义。下一节将探讨软件质量问题。

1.2 软件质量

当然，我们希望软件具有较高的质量。但是高质量的含义是什么呢？在通常情况下，有许多质量特征需要考虑。图 1.1 列出了高质量软件需要考虑的几个方面。

质量特征	描述
正确性	软件遵守特定需求的程度
可靠性	软件故障的频度和危险度
健壮性	妥善地处理错误情况的能力
可用性	用户用软件学习和执行任务的容易程度
可维护性	能够修改软件的容易程度
可重用性	软件组件在开发其他软件系统时被重用的容易度
可移植性	软件组件在多个计算机环境中使用的容易度
有效性	软件在不浪费资源的情况下实现目标的程度

图 1.1 软件质量的特征

1.2.1 正确性

正确性的概念可以追溯到解决方案的最初目标。在开发过程的每一步，我们需要确保按需求规格说明中规定的那样解决问题。如果软件不能解决正确的问题，则质量的其他方面几乎是毫无意义的。

正确性也隐含着解决方案产生正确的结果。这一概念绝非仅仅执行某种精度的数值计算。软件还应以一种良好组织的、令人看上去感到赏心悦目的方式显示图形和用户界面组件；它还应当产生恰当的措词和拼写正确的文本输出（包括错误消息）。

1.2.2 可靠性

如果你曾经尝试通过电子方式访问银行账号，但是不能够如愿以偿，或者曾经由于正在使用的软件或者硬件的故障而丢了工作，应该已经熟悉了可靠性的概念。软件故障可以定义为在允许的操作条件下发生的任何不可接受的行为。我们可以计算可靠性的测量值，诸如平均故障间隔时间。可靠性还应考虑这样的事实，即某些故障比另一些故障更严重。最重要的是，万一发生故障，软件应当不会造成其他破坏。

关键概念：可靠的软件很少会发生故障，并且在真正发生时，故障的影响面也尽量最小化。

在某些情况下，可靠性事关人命。20 世纪 80 年代早期，有一种医疗设备称为 Therac 25，它的主要作用是根据医生通过特殊键盘所做的设定，放出一定量的射线。但是控制该设备的软件中存在一个错误，当医生通过键盘调整到某个非常特殊的值时，设备的内部装置会发生剧烈的变化，并且放出射线的剂量足以致命。由于这种错误发生的概率很低，以致于死了几个人以后才发现问题的根源。

在其他情况下，可靠性会造成财务上的影响。在 1998 年 11 月的一天，美国东部的整个 AT&T 网络发生故障，导致严重的通信中断。问题最终归咎到某个具体的软件错误。仅此一次故障，就使有关公司的年收入损失上百万美元。

1.2.3 健壮性

可靠性与一个系统的健壮程度有关，一个健壮的系统处理问题时很平稳。例如，如果