

高等学校21世纪教材

GAODENG XUEXIAO 21 SHIJI JIAOCAI

# 软件工程

(第二版)

● 张海藩 编著



人民邮电出版社

POSTS & TELECOM PRESS

高等学校 21 世纪教材

# 软 件 工 程

(第 二 版)

张海藩 编著

人民邮电出版社

## 图书在版编目(CIP)数据

软件工程/张海藩编著:—2版.—北京:人民邮电出版社,2006.1

ISBN 7-115-14070-7

I. 软... II. 张... III. 软件工程—高等学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2005)第 114823 号

## 内 容 提 要

本书是《软件工程》的第二版。

本书由五篇共 16 章构成,第一篇讲述软件工程与软件过程;第二篇讲述结构化分析、设计与实现;第三篇讲述面向对象的概念、模型、分析、设计与实现;第四篇讲述软件项目的计划、组织和控制,并介绍了一些相关的国际标准;第五篇讲述形式化方法、统一建模语言 UML 和软件重用。

本书内容新颖、实例丰富,可作为高等院校“软件工程”课程的教材或教学参考书,也可供软件工程师、软件项目管理者和应用软件的开发人员阅读参考。

### 高等学校 21 世纪教材 软件工程(第二版)

- 
- ◆ 编 著 张海藩  
责任编辑 潘春燕
  - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
北京艺辉印刷有限公司印刷  
新华书店总店北京发行所经销
  - ◆ 开本:787×1092 1/16  
印张:23.75  
字数:560千字  
印数:89 101-94 100册
- 2006年1月第2版  
2006年1月北京第14次印刷

ISBN 7-115-14070-7/TP·5013

定价:31.00元

读者服务热线:(010)67170985 印装质量热线:(010)67129223

## 丛书前言

当今世界,科学技术突飞猛进,知识经济已见端倪,国际竞争日趋激烈。教育在综合国力的形成中处于基础地位,国力的强弱将越来越取决于劳动者的素质,取决于各类人才的质量和数量,这对于培养和造就我国 21 世纪的一代新人提出了更加迫切的要求。21 世纪初,我国高等教育呈快速发展的势头。教材是体现教学内容和教学方法的知识载体,是进行教学的基本工具,也是深化教育教学改革、全面推进素质教育、培养创新人才的重要保证。因此,高等教育教材建设必须有一个与之相适应的快速发展。

随着计算机软硬件的不断升级换代,计算机教学内容也随之更新,尤其随着教育部“高等教育面向 21 世纪教育内容与课程体系改革”计划的实施,对教材也提出了新的要求。为此我们聘请了国内高校计算机教学方面知名的专家教授,精心策划编写了这套“高等学校 21 世纪教材”。

为真正实施精品战略,组织编写好这套教材,我们在国内高校做了系统、详细的调查,对教育部制订的教育计划做了认真的研究,还对国内外已出版的教材做了理性的分析,确立了依托国家教育计划、传播先进教学理念、为培养符合社会需要的高素质创新型人才服务的宗旨。

在本套教材的策划过程中,我们多次组织了由专家及高校一线教师参加的研讨会,对现有比较出色的教材的特点及优点进行了分析,博采众长,力求实现教材权威性与实用性的完美结合。

本套教材有如下特点。

1. 考虑到全国普通高等院校学生的知识、能力、素质的特点和实际教学情况,在编写教材时把重点放在基本理论、基础知识、基本技能与方法上。
2. 紧密结合当前技术的新发展,在阐述理论知识的同时侧重实用性。
3. 力求在概念和原理的讲述上严格、准确、精练,理论适中,实例丰富,写作风格上深入浅出,图文并茂,便于学生学习。
4. 为适应当前高校课程种类多、课时数要压缩的教学特点,教材不仅篇幅有很大的压缩,而且均配有电子教案,以满足现代教学新特点的需要,做到易教易学。
5. 所选作者均是国内有丰富教学实践经验的知名专家、教授,所编教材具有较高的权威性。

教育的改革将不会停止,教材也将会不断推陈出新。目前本套教材即将推出,将接受广大教学第一线教师的检验。

由于我们的水平和经验有限,这批教材在编审、出版工作中还存在不少缺点和不足,希望使用本套教材的学校师生和广大读者提出批评和建议,以便改进我们的工作,使教材质量不断提高。

## 第一版编者的话

软件工程是指导计算机软件开发与维护的工程学科,它采用工程的概念、原理、技术和方法来开发与维护软件,把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来,以便经济地开发出高质量的软件并有效地维护它。经过 30 多年的研究与发展,软件工程正逐步走向成熟,但是,很多个人和公司仍然在随意地开发软件,许多软件专业的学生甚至软件工程师还没有掌握现代化的软件开发方法,以致我们生产的软件仍然存在大量的质量问题。因此认真学习并在实际工作中正确地运用软件工程,是摆在我们面前的一项十分迫切的任务。

编者从事软件工程教学工作近 20 年,亲身体会到教材对教师和学生的重要性。此次应邀编写《软件工程》一书,深感责任重大,因此,编写此书前从各方面都做了充分的准备。

本书由五篇共 16 章构成,按专题安排,便于授课教师和读者按需取用。

第一篇软件工程与软件过程,全面、概括地介绍软件工程这门学科及典型的软件过程模型。

第二篇传统方法学,分别讲述了结构化分析、结构化设计和结构化实现的概念、原理、准则、技术和方法。

第三篇面向对象方法学,分别讲述了面向对象的概念与模型、面向对象分析、面向对象设计和面向对象实现。面向对象方法学是本书重点讲述的内容,因此本篇在讲述面向对象分析和面向对象设计的过程中不仅始终结合了 ATM 系统的例子,而且还单独讲述了电梯系统的面向对象分析过程及 C++ 类库管理系统的面向对象分析与设计过程。这些完整的实例对读者深入、具体地理解面向对象方法学很有帮助,也是上机实习的好材料。

第四篇软件项目管理,分别讲述软件项目的计划、组织和控制,并简要地介绍了一些与软件项目管理有关的国际标准,供读者在实际工作中参考、借鉴。

第五篇高级课题,简要地介绍软件工程研究的一些新成果,讲述了软件工程的形式化方法、面向对象的统一建模语言 UML 和软件重用。

在本书的编写过程中,蔡英协助编者查找了一些资料,张劲松和张展新用 VC++ 实现了书中讲述的 C++ 类库管理系统,张雯和张杰完成了本书的一些具体工作,谨在此向他们表示感谢。

编者  
2002 年 2 月

## 第二版编者的话

本书第一版自 2002 年 3 月出版以来,深受广大读者欢迎,许多高校用它作为“软件工程概论”课的教材,累计销售约 8 万册。为了反映最近 3 年来软件工程的<sub>发展现状,更好地为广大读者服务,编者对第一版内容作了认真修改,形成了第二版。</sub>

在保持原书结构和篇幅基本不变的前提下,第二版主要对原书内容作了下述修改:

(1)更新了一些较陈旧的内容。例如,用 COCOMO2 模型取代了 COCOMO 模型。

(2)增加了一些较重要或较新颖的内容。例如,增加了结构化分析实例、面向对象程序设计实例、估算软件开发时间的方法等。

(3)用统一建模语言(UML)的概念与符号,重新改写了讲述面向对象方法学的第三篇。

此外,还对原书中一些具体内容作了修改或更新,对文字叙述作了进一步的加工和润色。

倪宁对第二版的内容提出了许多有益的建议,张杰和张雯为第二版的编写作了大量辅助性工作。谨在此向他们表示感谢!

编者  
2005 年 8 月

# 目 录

## 第一篇 软件工程与软件过程

|                         |    |
|-------------------------|----|
| <b>第 1 章 软件工程</b> ..... | 1  |
| 1.1 软件危机 .....          | 1  |
| 1.1.1 计算机系统的发展历程 .....  | 1  |
| 1.1.2 软件危机的含义 .....     | 2  |
| 1.1.3 产生软件危机的原因 .....   | 3  |
| 1.1.4 消除软件危机的途径 .....   | 5  |
| 1.2 软件工程 .....          | 6  |
| 1.2.1 什么是软件工程 .....     | 6  |
| 1.2.2 软件工程的基本原理 .....   | 6  |
| 1.2.3 软件工程方法学 .....     | 8  |
| 1.3 小结.....             | 10 |
| 习题 .....                | 11 |
| <b>第 2 章 软件过程</b> ..... | 13 |
| 2.1 软件生命周期的基本任务.....    | 13 |
| 2.2 瀑布模型.....           | 16 |
| 2.3 快速原型模型.....         | 18 |
| 2.4 增量模型.....           | 19 |
| 2.5 螺旋模型.....           | 20 |
| 2.6 喷泉模型.....           | 22 |
| 2.7 小结.....             | 23 |
| 习题 .....                | 24 |

## 第二篇 传统方法学

|                          |    |
|--------------------------|----|
| <b>第 3 章 结构化分析</b> ..... | 25 |
| 3.1 概述.....              | 25 |
| 3.2 与用户通信的技术.....        | 26 |
| 3.2.1 访谈.....            | 26 |
| 3.2.2 简易的应用规格说明技术.....   | 27 |
| 3.2.3 软件原型.....          | 28 |
| 3.3 分析建模与规格说明.....       | 29 |
| 3.3.1 分析建模.....          | 29 |

|            |                 |           |
|------------|-----------------|-----------|
| 3.3.2      | 软件需求规格说明        | 29        |
| 3.4        | 实体—关系图          | 31        |
| 3.4.1      | 数据对象            | 31        |
| 3.4.2      | 属性              | 31        |
| 3.4.3      | 关系              | 32        |
| 3.4.4      | 实体—关系图的符号       | 32        |
| 3.5        | 数据流图            | 33        |
| 3.5.1      | 数据流图符号          | 33        |
| 3.5.2      | 例子              | 34        |
| 3.5.3      | 命名              | 37        |
| 3.6        | 状态转换图           | 37        |
| 3.6.1      | 状态              | 38        |
| 3.6.2      | 事件              | 38        |
| 3.6.3      | 符号              | 38        |
| 3.6.4      | 例子              | 39        |
| 3.7        | 数据字典            | 40        |
| 3.8        | 结构化分析实例         | 41        |
| 3.8.1      | 问题陈述            | 42        |
| 3.8.2      | 问题定义            | 42        |
| 3.8.3      | 可行性研究           | 43        |
| 3.8.4      | 需求分析            | 48        |
| 3.9        | 小结              | 53        |
|            | 习题              | 55        |
| <b>第4章</b> | <b>结构化设计</b>    | <b>56</b> |
| 4.1        | 结构化设计与结构化分析的关系  | 56        |
| 4.2        | 软件设计的概念和原理      | 57        |
| 4.2.1      | 模块化             | 57        |
| 4.2.2      | 抽象              | 59        |
| 4.2.3      | 逐步求精            | 59        |
| 4.2.4      | 信息隐藏            | 60        |
| 4.3        | 模块独立            | 61        |
| 4.3.1      | 耦合              | 61        |
| 4.3.2      | 内聚              | 62        |
| 4.4        | 启发规则            | 63        |
| 4.4.1      | 改进软件结构提高模块独立性   | 63        |
| 4.4.2      | 模块规模应该适中        | 64        |
| 4.4.3      | 深度、宽度、扇出和扇入都应适当 | 64        |
| 4.4.4      | 模块的作用域应该在控制域之内  | 64        |
| 4.4.5      | 力争降低模块接口的复杂程度   | 65        |



|                     |           |
|---------------------|-----------|
| 4.4.6 设计单入口单出口的模块   | 65        |
| 4.4.7 模块功能应该可以预测    | 65        |
| 4.5 表示软件结构的图形工具     | 65        |
| 4.5.1 层次图和 HIPO 图   | 65        |
| 4.5.2 结构图           | 67        |
| 4.6 面向数据流的设计方法      | 68        |
| 4.6.1 概念            | 68        |
| 4.6.2 变换分析          | 69        |
| 4.6.3 事务分析          | 75        |
| 4.6.4 设计优化          | 76        |
| 4.7 人机界面设计          | 77        |
| 4.7.1 人机界面设计问题      | 77        |
| 4.7.2 人机界面设计过程      | 78        |
| 4.7.3 界面设计指南        | 79        |
| 4.8 过程设计            | 81        |
| 4.9 过程设计的工具         | 83        |
| 4.9.1 程序流程图         | 83        |
| 4.9.2 盒图(N-S图)      | 84        |
| 4.9.3 PAD图          | 84        |
| 4.9.4 判定表           | 86        |
| 4.9.5 判定树           | 87        |
| 4.9.6 过程设计语言(PDL)   | 87        |
| 4.10 面向数据结构的设计方法    | 88        |
| 4.10.1 Jackson图     | 88        |
| 4.10.2 改进的 Jackson图 | 89        |
| 4.10.3 Jackson方法    | 89        |
| 4.11 小结             | 94        |
| 习题                  | 95        |
| <b>第5章 结构化实现</b>    | <b>98</b> |
| 5.1 编码              | 98        |
| 5.1.1 选择程序设计语言      | 98        |
| 5.1.2 编码风格          | 100       |
| 5.2 软件测试基础          | 102       |
| 5.2.1 测试目标          | 102       |
| 5.2.2 黑盒测试和白盒测试     | 103       |
| 5.2.3 测试准则          | 103       |
| 5.2.4 流图            | 103       |
| 5.3 逻辑覆盖            | 106       |
| 5.4 控制结构测试          | 108       |

|       |                    |     |
|-------|--------------------|-----|
| 5.4.1 | 基本路径测试 .....       | 108 |
| 5.4.2 | 条件测试 .....         | 111 |
| 5.4.3 | 数据流测试 .....        | 113 |
| 5.4.4 | 循环测试 .....         | 114 |
| 5.5   | 黑盒测试技术 .....       | 115 |
| 5.5.1 | 等价划分 .....         | 116 |
| 5.5.2 | 边界值分析 .....        | 118 |
| 5.5.3 | 错误推测 .....         | 119 |
| 5.6   | 测试策略 .....         | 120 |
| 5.6.1 | 测试步骤 .....         | 120 |
| 5.6.2 | 单元测试 .....         | 121 |
| 5.6.3 | 集成测试 .....         | 123 |
| 5.6.4 | 确认测试 .....         | 126 |
| 5.7   | 调试 .....           | 127 |
| 5.7.1 | 调试过程 .....         | 127 |
| 5.7.2 | 调试途径 .....         | 128 |
| 5.8   | 软件可靠性 .....        | 130 |
| 5.8.1 | 基本概念 .....         | 130 |
| 5.8.2 | 估算平均无故障时间的方法 ..... | 131 |
| 5.9   | 小结 .....           | 132 |
|       | 习题 .....           | 133 |

### 第三篇 面向对象方法学

|            |                        |            |
|------------|------------------------|------------|
| <b>第6章</b> | <b>面向对象方法学导论 .....</b> | <b>139</b> |
| 6.1        | 面向对象程序设计实例 .....       | 139        |
| 6.1.1      | 用对象分解取代功能分解 .....      | 139        |
| 6.1.2      | 设计类等级 .....            | 141        |
| 6.1.3      | 定义属性和服务 .....          | 143        |
| 6.1.4      | 用C++语言实现 .....         | 144        |
| 6.2        | 面向对象方法学概述 .....        | 147        |
| 6.2.1      | 面向对象方法学的要点 .....       | 147        |
| 6.2.2      | 面向对象的软件过程 .....        | 149        |
| 6.3        | 面向对象方法学的主要优点 .....     | 150        |
| 6.4        | 面向对象的概念 .....          | 153        |
| 6.4.1      | 对象 .....               | 153        |
| 6.4.2      | 其他概念 .....             | 156        |
| 6.5        | 面向对象建模 .....           | 159        |
| 6.6        | 对象模型 .....             | 160        |
| 6.6.1      | 表示类的符号 .....           | 160        |

|                            |            |
|----------------------------|------------|
| 6.6.2 表示关系的符号 .....        | 161        |
| 6.7 动态模型 .....             | 164        |
| 6.8 功能模型 .....             | 165        |
| 6.9 三种模型之间的关系 .....        | 165        |
| 6.10 小结 .....              | 166        |
| 习题 .....                   | 166        |
| <b>第7章 面向对象分析</b> .....    | <b>168</b> |
| 7.1 分析过程 .....             | 168        |
| 7.1.1 概述 .....             | 168        |
| 7.1.2 三个子模型与五个层次 .....     | 169        |
| 7.2 需求陈述 .....             | 170        |
| 7.2.1 书写要点 .....           | 170        |
| 7.2.2 例子 .....             | 171        |
| 7.3 建立对象模型 .....           | 172        |
| 7.3.1 确定类与对象 .....         | 172        |
| 7.3.2 确定关联 .....           | 174        |
| 7.3.3 划分主题 .....           | 177        |
| 7.3.4 确定属性 .....           | 177        |
| 7.3.5 识别继承关系 .....         | 178        |
| 7.3.6 反复修改 .....           | 179        |
| 7.4 建立动态模型 .....           | 182        |
| 7.4.1 编写脚本 .....           | 182        |
| 7.4.2 设想用户界面 .....         | 183        |
| 7.4.3 画事件跟踪图 .....         | 184        |
| 7.4.4 画状态图 .....           | 184        |
| 7.4.5 审查动态模型 .....         | 186        |
| 7.5 建立功能模型 .....           | 187        |
| 7.5.1 画出基本系统模型图 .....      | 187        |
| 7.5.2 画出功能级数据流图 .....      | 187        |
| 7.5.3 描述处理框功能 .....        | 188        |
| 7.6 定义服务 .....             | 189        |
| 7.6.1 常规行为 .....           | 189        |
| 7.6.2 从事件导出的操作 .....       | 189        |
| 7.6.3 与数据流图中处理框对应的操作 ..... | 189        |
| 7.6.4 利用继承减少冗余操作 .....     | 189        |
| 7.7 面向对象分析实例 .....         | 190        |
| 7.7.1 需求陈述 .....           | 190        |
| 7.7.2 建立对象模型 .....         | 190        |
| 7.7.3 建立动态模型 .....         | 191        |

|                           |            |
|---------------------------|------------|
| 7.7.4 建立功能模型 .....        | 193        |
| 7.7.5 进一步完善 .....         | 194        |
| 7.8 小结 .....              | 194        |
| 习题 .....                  | 195        |
| <b>第8章 面向对象设计</b> .....   | <b>197</b> |
| 8.1 面向对象设计的准则 .....       | 197        |
| 8.1.1 模块化 .....           | 198        |
| 8.1.2 抽象 .....            | 198        |
| 8.1.3 信息隐藏 .....          | 198        |
| 8.1.4 弱耦合 .....           | 198        |
| 8.1.5 强内聚 .....           | 199        |
| 8.1.6 可重用 .....           | 199        |
| 8.2 启发规则 .....            | 199        |
| 8.2.1 设计结果应该清晰易懂 .....    | 200        |
| 8.2.2 一般—特殊结构的深度应适当 ..... | 200        |
| 8.2.3 设计简单的类 .....        | 200        |
| 8.2.4 使用简单的协议 .....       | 201        |
| 8.2.5 使用简单的服务 .....       | 201        |
| 8.2.6 把设计变动减至最小 .....     | 201        |
| 8.3 系统分解 .....            | 201        |
| 8.3.1 子系统之间的两种交互方式 .....  | 202        |
| 8.3.2 组织系统的两种方案 .....     | 202        |
| 8.3.3 设计系统的拓扑结构 .....     | 203        |
| 8.4 设计问题域子系统 .....        | 203        |
| 8.4.1 调整需求 .....          | 204        |
| 8.4.2 重用已有的类 .....        | 204        |
| 8.4.3 把问题域类组合在一起 .....    | 204        |
| 8.4.4 增添一般化类以建立协议 .....   | 204        |
| 8.4.5 ATM系统之例 .....       | 204        |
| 8.5 设计人一机交互子系统 .....      | 205        |
| 8.5.1 设计人一机交互界面的准则 .....  | 205        |
| 8.5.2 设计人一机交互子系统的策略 ..... | 206        |
| 8.6 设计任务管理子系统 .....       | 207        |
| 8.6.1 分析并发性 .....         | 208        |
| 8.6.2 设计任务管理子系统 .....     | 208        |
| 8.7 设计数据管理子系统 .....       | 209        |
| 8.7.1 选择数据存储管理模式 .....    | 209        |
| 8.7.2 设计数据管理子系统 .....     | 210        |
| 8.7.3 例子 .....            | 212        |

|                         |            |
|-------------------------|------------|
| 8.8 设计类中的服务 .....       | 212        |
| 8.8.1 确定类中应有的服务 .....   | 212        |
| 8.8.2 设计实现服务的方法 .....   | 213        |
| 8.9 设计关联 .....          | 214        |
| 8.9.1 关联的遍历 .....       | 214        |
| 8.9.2 实现单向关联 .....      | 214        |
| 8.9.3 实现双向关联 .....      | 215        |
| 8.9.4 关联对象的实现方法 .....   | 215        |
| 8.10 设计优化 .....         | 215        |
| 8.10.1 确定优先级 .....      | 215        |
| 8.10.2 提高效率的几项技术 .....  | 216        |
| 8.10.3 调整继承关系 .....     | 217        |
| 8.11 面向对象分析与设计实例 .....  | 218        |
| 8.11.1 面向对象分析 .....     | 219        |
| 8.11.2 面向对象设计 .....     | 221        |
| 8.12 小结 .....           | 225        |
| 习题 .....                | 226        |
| <b>第9章 面向对象实现 .....</b> | <b>227</b> |
| 9.1 程序设计语言 .....        | 227        |
| 9.1.1 面向对象语言的优点 .....   | 227        |
| 9.1.2 面向对象语言的技术特点 ..... | 228        |
| 9.1.3 选择面向对象语言 .....    | 231        |
| 9.2 程序设计风格 .....        | 232        |
| 9.2.1 提高可重用性 .....      | 232        |
| 9.2.2 提高可扩充性 .....      | 234        |
| 9.2.3 提高健壮性 .....       | 234        |
| 9.3 测试策略 .....          | 235        |
| 9.3.1 面向对象的单元测试 .....   | 235        |
| 9.3.2 面向对象的集成测试 .....   | 235        |
| 9.3.3 面向对象的确认测试 .....   | 236        |
| 9.4 设计测试用例 .....        | 236        |
| 9.4.1 测试类的方法 .....      | 236        |
| 9.4.2 集成测试方法 .....      | 238        |
| 9.5 小结 .....            | 240        |
| 习题 .....                | 240        |
| <b>第四篇 软件项目管理</b>       |            |
| <b>第10章 计划 .....</b>    | <b>242</b> |
| 10.1 度量软件规模 .....       | 242        |

|               |            |     |
|---------------|------------|-----|
| 10.1.1        | 代码行技术      | 242 |
| 10.1.2        | 功能点技术      | 243 |
| 10.2          | 工作量估算      | 245 |
| 10.2.1        | 静态单变量模型    | 245 |
| 10.2.2        | 动态多变量模型    | 245 |
| 10.2.3        | COCOMO2 模型 | 246 |
| 10.3          | 进度计划       | 248 |
| 10.3.1        | 基本原则       | 249 |
| 10.3.2        | 估算软件开发时间   | 250 |
| 10.3.3        | Gantt 图    | 251 |
| 10.3.4        | 工程网络       | 252 |
| 10.3.5        | 估算进度       | 253 |
| 10.3.6        | 关键路径       | 254 |
| 10.3.7        | 机动时间       | 255 |
| 10.4          | 小结         | 256 |
|               | 习题         | 257 |
| <b>第 11 章</b> | <b>组织</b>  | 259 |
| 11.1          | 民主制程序员组    | 259 |
| 11.2          | 主程序员组      | 260 |
| 11.3          | 现代程序员组     | 262 |
| 11.4          | 软件项目组      | 263 |
| 11.4.1        | 三种组织方式     | 264 |
| 11.4.2        | 四种组织范型     | 265 |
| 11.5          | 小结         | 266 |
|               | 习题         | 266 |
| <b>第 12 章</b> | <b>控制</b>  | 267 |
| 12.1          | 风险管理       | 267 |
| 12.1.1        | 软件风险分类     | 267 |
| 12.1.2        | 风险识别       | 268 |
| 12.1.3        | 风险预测       | 272 |
| 12.1.4        | 处理风险的策略    | 274 |
| 12.2          | 质量保证       | 275 |
| 12.2.1        | 软件质量       | 275 |
| 12.2.2        | 软件质量保证措施   | 276 |
| 12.3          | 配置管理       | 279 |
| 12.3.1        | 软件配置       | 279 |
| 12.3.2        | 软件配置管理过程   | 281 |
| 12.4          | 小结         | 286 |
|               | 习题         | 286 |

|                                      |     |
|--------------------------------------|-----|
| <b>第 13 章 国际标准</b> .....             | 288 |
| 13.1 IEEE 1058.1 软件项目管理计划标准 .....    | 288 |
| 13.1.1 软件项目管理计划的组成 .....             | 288 |
| 13.1.2 IEEE 软件项目管理计划 .....           | 289 |
| 13.2 ISO9000 质量标准 .....              | 291 |
| 13.2.1 基本思想 .....                    | 291 |
| 13.2.2 ISO9000—3 标准 .....            | 292 |
| 13.3 ISO/IEC 12207 软件生命周期过程标准 .....  | 294 |
| 13.3.1 概述 .....                      | 294 |
| 13.3.2 软件生命周期过程 .....                | 295 |
| 13.4 ISO/IEC TR 15504 软件过程评估标准 ..... | 298 |
| 13.4.1 概述 .....                      | 299 |
| 13.4.2 标准的结构 .....                   | 299 |
| 13.5 能力成熟度模型 .....                   | 300 |
| 13.5.1 能力成熟度模型的结构 .....              | 301 |
| 13.5.2 能力成熟度等级 .....                 | 302 |
| 13.5.3 关键过程域 .....                   | 303 |
| 13.5.4 应用 CMM .....                  | 304 |
| 13.6 小结 .....                        | 304 |
| 习题 .....                             | 305 |

## 第五篇 高级课题

|                           |     |
|---------------------------|-----|
| <b>第 14 章 形式化方法</b> ..... | 306 |
| 14.1 概述 .....             | 306 |
| 14.1.1 非形式化方法的缺点 .....    | 306 |
| 14.1.2 软件开发过程中的数学 .....   | 307 |
| 14.1.3 应用形式化方法的准则 .....   | 307 |
| 14.2 有穷状态机 .....          | 308 |
| 14.2.1 基本概念 .....         | 308 |
| 14.2.2 电梯问题 .....         | 309 |
| 14.2.3 评论 .....           | 312 |
| 14.3 Petri 网 .....        | 312 |
| 14.3.1 基本概念 .....         | 312 |
| 14.3.2 应用实例 .....         | 314 |
| 14.4 Z 语言 .....           | 315 |
| 14.4.1 简介 .....           | 315 |
| 14.4.2 评论 .....           | 317 |
| 14.5 小结 .....             | 318 |
| 习题 .....                  | 318 |

|                                |     |
|--------------------------------|-----|
| <b>第 15 章 统一建模语言 UML</b> ..... | 320 |
| 15.1 概述.....                   | 320 |
| 15.1.1 UML 的产生和发展 .....        | 320 |
| 15.1.2 UML 的结构 .....           | 321 |
| 15.1.3 UML 的图 .....            | 322 |
| 15.1.4 UML 的应用领域 .....         | 323 |
| 15.2 静态建模机制.....               | 324 |
| 15.2.1 用例图.....                | 324 |
| 15.2.2 类图和对象图.....             | 327 |
| 15.3 动态建模机制.....               | 336 |
| 15.3.1 消息.....                 | 336 |
| 15.3.2 状态图.....                | 336 |
| 15.3.3 顺序图.....                | 337 |
| 15.3.4 协作图.....                | 339 |
| 15.3.5 活动图.....                | 340 |
| 15.4 描述物理架构的机制.....            | 341 |
| 15.4.1 逻辑架构和物理架构.....          | 341 |
| 15.4.2 构件图.....                | 341 |
| 15.4.3 配置图.....                | 342 |
| 15.5 使用和扩展 UML .....           | 343 |
| 15.5.1 使用 UML 的准则 .....        | 343 |
| 15.5.2 扩展 UML 的机制 .....        | 344 |
| 15.6 小结.....                   | 345 |
| 习题.....                        | 345 |
| <b>第 16 章 软件重用</b> .....       | 346 |
| 16.1 可重用的软件成分.....             | 346 |
| 16.2 软件重用过程.....               | 347 |
| 16.2.1 构件组装模型.....             | 347 |
| 16.2.2 类构件.....                | 348 |
| 16.2.3 重用过程模型.....             | 349 |
| 16.3 领域工程.....                 | 350 |
| 16.3.1 分析过程.....               | 350 |
| 16.3.2 领域特征.....               | 351 |
| 16.3.3 结构建模和结构点.....           | 352 |
| 16.4 开发可重用的构件.....             | 352 |
| 16.4.1 为了重用的分析与设计.....         | 352 |
| 16.4.2 基于构件的开发.....            | 353 |
| 16.5 分类和检索构件.....              | 354 |
| 16.5.1 描述可重用的构件.....           | 355 |



|                   |            |
|-------------------|------------|
| 16.5.2 重用环境.....  | 357        |
| 16.6 软件重用的效益..... | 357        |
| 16.7 小结.....      | 358        |
| 习题.....           | 359        |
| <b>参考文献</b> ..... | <b>360</b> |