

软件工程与计算（卷二）

软件开发的 技术基础

骆斌 主编 丁二玉 刘钦 编著

*S*oftware Engineering
and Computing (Volume II)
Fundamentals of Software
Development Technology



机械工业出版社
China Machine Press

. 013033021

高等院校软件工程专业规划教材

TP311.5-43

182

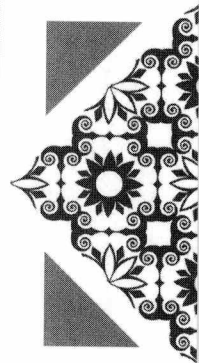
V2

软件工程与计算（卷二）

软件开发的 技术基础

骆斌 主编 丁二玉 刘钦 编著

*S*oftware Engineering
and Computing (Volume II)
**Fundamentals of Software
Development Technology**



北航

C1640747

TP311.5-43

182

V2



机械工业出版社
China Machine Press

100880810

图书在版编目 (CIP) 数据

软件工程与计算 (卷二): 软件开发的技术基础 / 骆斌主编. —北京: 机械工业出版社, 2012.12
(高等院校软件工程专业规划教材)
ISBN 978-7-111-40750-8

I. 软… II. 骆… III. ①软件工程—高等学校—教材 ②软件开发—高等学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2013) 第 048868 号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

作为国家精品课程“软件工程与计算”系列课程的第二门课程配套教材, 本书以经典软件工程方法与技术为主线, 软件开发技术与程序设计知识为教学重点, 培养学生简单小组级别、中小规模软件系统的软件开发能力。

全书主要分为六部分。第一部分介绍软件工程的基本框架。第二部分介绍项目启动阶段的知识。第三部分介绍软件需求开发的基础知识, 包括软件需求工程的概要、软件需求的内涵、常见的需求分析方法、软件需求文档。第四部分首先介绍软件设计的基础概念, 之后沿着设计过程和设计技术两条主线, 深入描述软件设计的相关知识。第五部分介绍软件构造、测试、移交与维护等软件开发的下游工程的基础知识。第六部分是对第一部分的延续, 通过总结性回顾, 进一步加深读者对软件工程的理解。

本书可作为高等院校软件工程、计算机及相关专业本科生软件工程课程的教材, 也可作为从事软件开发的相关技术人员的参考书。

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 朱秀英

北京京师印务有限公司印刷

2012 年 12 月第 1 版第 1 次印刷

185mm × 260mm · 28.5 印张

标准书号: ISBN 978-7-111-40750-8

定 价: 55.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com

软件工程专业教育源于软件产业界的现实人才需求和计算学科教程 CC1991/2001/2005 的不断推动, CC1991 明确提出计算机科学学科教学计划已经不适应产业需求, 应将其上升到计算学科教学计划予以考虑, CC2001 提出了计算机科学、计算机工程、软件工程、信息系统 4 个子学科, CC2005 增加了信息技术子学科, 并发布了正式版的软件工程等子学科教学计划建议。我国的软件工程本科教育启动于 2002 年, 与国际基本同步, 目前该专业招生人数已经进入国内高校本科专业前十位, 软件工程专业课程体系建设与教材建设是摆在中国软件工程教育工作者面前的一个重要任务。

国际软件工程学科教程 CC-SE2004 建议, 软件工程专业教学计划的技术课程包括初级课程、中级课程、高级课程和领域相关课程。

- 初级课程。包括离散数学、数据结构与算法两门公共课程, 另三门课程可以组织成计算机科学优先方案(程序设计基础、面向对象方法、软件工程导论)和软件工程优先方案(软件工程与计算概论 / 软件工程与计算 II / 软件工程与计算 III)。
- 中级课程。覆盖计算机硬件、操作系统、网络、数据库以及其他必备的计算机硬件与计算机系统基本知识, 课程总数与计算机专业相比应大幅度缩减。
- 高级课程。六门课程, 覆盖软件需求、体系结构、设计、构造、测试、质量、过程、管理和人机交互等。
- 领域相关课程。与具体应用领域相关的选修课程, 所有学校应结合办学特色开设。

CC-SE2004 的实践难点在于: 如何把计算机专业的一门软件工程课程按照教学目标有效拆分成初级课程和六门高级课程? 如何裁剪与求精计算机硬件与系统课程? 如何在专业教学初期引入软件工程观念, 并将其在教学中与程序设计、软件职业、团队交流沟通相结合?

南京大学一直致力于基于 CC-SE2004 规范的软件工程教学实践与创新, 在专业教学早期注重培养学生的软件工程观与计算机系统观, 按照软件系统由小及大的线索从一年级开始组织软件工程类课程。具体做法是: 在求精计算机硬件与系统课程的基础上, 融合软件工程基础、程序设计、职业团队等知识实践的“软件工程与计算”系列课程, 通过案例教授中小规模软件系统构建; 围绕大中型软件系统构建知识分领域, 组织软件工程高级课程; 围绕软件工程应用领域, 建设领域相关课程。南京大学的“软件工程与计算”、“计算系统基础”和“操作系统”是国家级精品课程, “软件需求工程”、“软件过程与管理”是教育部-IBM 精品课程, 软件工程专业工程化实践教学体系和人才培养体系分别获得第五届与第六届高等教育

国家级教学成果奖。

此次集中出版的五本教材是软件工程专业课程建设工作的第二波，包括《软件工程与计算卷》的全部三分册（《软件开发的编程基础》、《软件开发的技术基础》、《团队与软件开发实践》）和《软件工程高级技术卷》的《人机交互——软件工程视角》与《软件过程与管理》。其中《软件工程与计算卷》围绕个人小规模软件系统、小组中小规模软件系统和模拟团队级中规模软件产品构建实践了 CC-SE2004 软件工程优先的基础课程方案；《人机交互——软件工程视角》是为数不多的“人机交互的软件工程方法”教材；《软件过程与管理》则结合了个人级、小组级、组织级的软件过程。这五本教材在教学内容组织上立意较新，在国际国内可供参考的同类教科书很少，代表了我们对软件工程专业新课程教学的理解与探索，因此难免存在瑕疵与谬误，欢迎各位读者批评指正。

本教材系列得到教育部“质量工程”之软件工程主干课程国家级教学团队、软件工程国家级特色专业、软件工程国家级人才培养模式创新实验区、教育部“十二五本科教学工程”之软件工程国家级专业综合改革试点、软件工程国家级工程实践教育基地、计算机科学与软件工程国家级实验教学示范中心，以及南京大学 985 项目和有关出版社的支持。在本教材系列的建设过程中，南京大学的张大良先生、陈道蓄先生、李宣东教授、赵志宏教授，以及国防科学技术大学、清华大学、中国科学院软件所、北京航空航天大学、浙江大学、上海交通大学、复旦大学的一些软件工程教育专家给出了大量宝贵意见。特此鸣谢！

南京大学软件学院

2012 年 10 月

如何在软件工程专业教育早期培养学生的工程观念，并为高阶课程提供合理的知识和技能基础是摆在软件工程教育者面前的一个重要问题。我们编写了《软件工程与计算》三卷本教材（《软件开发的编程基础》、《软件开发的技术基础》、《团队与软件开发实践》）作为软件工程本科专业入门课程教材，帮助学生以工程化方法构建中小规模软件系统的知识和技能，并为后继高阶课程的学习打下全面基础。

教学实施建议

在使用《软件工程与计算》三卷本作为教材时，应当注意本套教材并不是“程序设计基础”、“面向对象方法”、“软件工程导论”、“软件职业基础”和“团队交流动力学”等课程的简单对应。在教学方式上，“软件工程与计算”的教学应当围绕构建中小规模计算系统（软件）这一主线，体现程序设计、面向对象方法、软件工程技术、软件工程管理、软件职业基础、团队交流技术的教学融合。

- 在教学中结合软件系统构造，培养学生的软件工程观念与职业认知。
- 建立围绕计算系统示例逐次构建不同规模软件系统的教学主线，以软件产品构建示例组织教学活动，借助三个典型的软件开发过程模型（迭代式开发模型、瀑布模型、螺旋模型），从小规模系统向中规模系统构建实践逐步演进。同时应当围绕该教学主线，组织学生进行实践，在实践中学习知识并将知识运用融会贯通。
- 加强学生对软件工程制品和软件工程工具的全面认知，始终强调软件开发制品，而不是在分离课程中分别强调计算系统代码和软件工程文档。
- 强调学生的课后阅读，强化学生的自学习能力。工程标准、语言规范、工具使用、文档格式等材料更多地应该通过课后阅读（而不是课堂讲解）传授给学生，学生通过系统地阅读这些材料并进一步在实践中加以运用，来提高自学习能力。
- 在教学执行过程中还应该考虑对知识产权的尊重，这本身是软件职业基础的一部分。

前驱课程

在完整使用《软件工程与计算》（三卷）作为软件工程专业入门课程教材使用时，考虑

到学生对于软件工程的理解难度，建议学生应当先修“计算系统基础”课程（也可以是计算机导论等课程）（下图方案一），使学生了解计算系统的分层构建方法和结构化程序设计基础。如果希望以本教材第一卷《软件工程与计算（卷一）：软件开发的编程基础》作为第一门专业课程教材，教师应当在课程中适当增加内容与课时，为学生建立起计算系统的基本概念并加强程序设计的教学时数（下图方案二），这样学生才能够更好地理解软件系统的构建。

建议“软件工程与计算”课程在大一下、大二上、大二下三个学期实施。

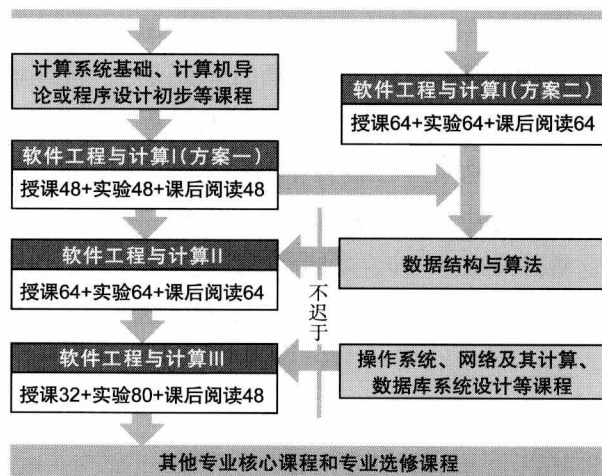
后继课程

本教材注重于中小规模计算系统（软件）构建中适用的软件工程方法和程序设计技术，按照“适与精”的原则组织软件工程与程序设计知识的教学内容。而软件工程学科知识的深度和全面性则应在后继课程中考虑：

- 那些没有被涉及的“系统全面”的软件工程知识与“适用于大规模系统”的软件工程方法，应按照领域组织在面向软件工程的专业核心课程群（例如软件需求、设计、构造、测试、质量、过程、管理等高阶软件工程课程）中。
- 其他程序设计类课程（数据结构与算法、软件设计、软件构造、软件架构等课程）讲解专门程序设计机制的使用。
- 适用于特定计算环境的软件工程高级方法和系统级应用程序设计接口应安排在“数据库系统设计”、“操作系统”、“网络及其计算”等面向计算环境的专业核心课程中。

课程的建议教学次序与建议教学课时数

软件工程专业或计算学科偏软件专业在实施“软件工程与计算”课程教学时，可以参照下图给出的教学顺序。



图《软件工程与计算》在专业教学中的建议执行次序与教学课时数

如上图所示，基于多年的教学实践和总结，我们建议在专业教学中实施“软件工程与计算”课程教学时采用如下执行次序和教学课时数：

- “软件工程与计算 I” 有两套教学执行方案：
 - “软件工程与计算 II” 可以按照方案一在执行“计算系统基础”先导课程的基础上执行，在大学一年级下学期开设，建议教学课时数为授课 48+ 实验 48+ 课后阅读 48。
 - “软件工程与计算 I” 也可以按照方案二作为专业入门课程，在一年级开设，建议教学课时数为授课 64+ 实验 64+ 课后阅读 64。
- “数据结构与算法” 在“软件工程与计算 I” 之后开设，“软件工程与计算 II” 在“数据结构与算法” 之后或同步开设，一般在二年级执行，建议教学课时数为授课 64+ 实验 64+ 课后阅读 64。
- “操作系统”、“网络及其计算”、“数据库系统设计” 在“数据结构与算法” 之后开设，“软件工程与计算 III” 与“操作系统”、“网络及其计算”、“数据库系统设计” 同步开设，一般在二年级下学期或三年级上学期执行，建议教学课时数为授课 32+ 实验 80+ 课后阅读 48。
- 其他课程在“软件工程与计算 III” 之后开设。

独立使用教材

本教材也可以独立使用，但应当注意以下事项。

《软件工程与计算（卷一）：软件开发的编程基础》：如果独立使用本书进行程序设计课程教学，那么需要容纳更多的程序设计知识的教学课时数，但是建议保持对调试、构建等与程序设计联系较为紧密的知识的教学以培养学生的实践能力。

《软件工程与计算（卷二）：软件开发的技术基础》：如果独立使用本书进行软件工程概论或者软件工程导论课程教学，那么可以适当弱化对详细设计和构造知识的教学，并补充过程与管理知识，强化软件需求与软件体系结构知识。

《软件工程与计算（卷三）：团队与软件开发实践》：如果独立使用本书进行软件工程实践课程教学，那么可以适当弱化课程的理论部分，补充技术回顾知识。

软件工程本科教育中的一个重要问题是如何在软件工程教育的开始阶段让学生建立工程观念，并为高阶课程提供合理的知识和技能基础。《软件工程与计算》(三卷)作为软件工程本科专业入门课程教材，帮助学生学习工程化构建中小型软件系统的知识和技能，并为后续高阶软件工程课程的学习打下全面的基础。

本书在写作过程中遵循了以下思路。

1. 围绕计算系统示例按照瀑布模型展开。本书围绕计算系统示例的开发构建，按照典型的瀑布式软件开发过程模型组织教学内容，详细描述了计算系统示例如何完整地逐步构建起来，以及所用到的软件工程开发技术。

2. 强调软件工程过程与软件开发技术的融合。本书融软件工程技术、程序设计技术、面向对象技术、软件工程管理、软件职业素质、团队合作交流等知识教学为一体。传统课程往往以技术主题为线索组织教学，容易使学生割裂理解各种软件开发技术。本课程以计算系统构建示例(计算系统示例与学生实践用例)为线索组织教学活动，让学生带着构建系统的问题，系统地学习知识并在适当的软件工程过程中融会贯通所学的软件开发技术。

3. 重点强调中小规模软件设计。本书的重点和难点是中小规模软件设计，希望读者通过本书的学习能够具有中小规模小组级软件开发构建的能力。而软件需求工程、大规模软件设计、软件构造、软件测试、软件过程与管理等高级知识和技能建议在后续高级课程中讲解。

4. 软件工程制品和软件工程工具的全面认知。软件工程制品不仅仅是代码，也不仅仅是文档，而是一系列紧密相关的软件产物，因此，本书反复强调学生对软件工程制品的认知和综合运用能力。同时，方法与工具是软件工程的两大支柱，因此，本书在传授综合知识和方法技术的同时，在计算系统示例讲解和配套学生实践用例实施过程中重视学生对软件工程工具、软件构造工具、项目管理工具等的使用。“工欲善其事，必先利其器”，本书推荐了一些主流的工具，但各校在执行过程中还应该考虑对知识产权的尊重，这本身也是软件职业基础的一部分。

本书由六个主要部分组成：

第一部分的基本目标是介绍软件工程的基本框架，使不熟悉软件工程的读者建立对软件工程的基本印象。

第二部分的基本目标是掌握项目启动阶段的知识，并能够实际开展相应活动。

第三部分的基本目标是介绍软件需求开发的基础知识，包括了解软件需求工程的概要，

理解软件需求的内涵，掌握常见的需求分析方法，能够编写简单的软件需求文档。

第四部分是全书的重点和难点，其基本目标是使读者掌握中小规模软件设计所需的相关技术。本部分首先介绍软件设计的基础概念，之后沿着设计过程和设计技术两条主线，深入描述软件设计的相关知识。其中主要包括软件设计的核心思想，设计模型，体系结构设计、人机交互设计和详细设计的过程，体系结构概念和风格，人机交互设计常用原则和技术，模块化与信息隐藏的思想，设计模式，以及如何编写软件设计描述文档。

第五部分的基本目标就是介绍下游工程的基础知识。软件构造、测试、移交与维护又被称为软件开发的下游工程。相比之下，上游工程更注重创造性，下游工程更注重将上游工程的结果进行成功实施。

第六部分是对第一部分的延续，基本目标是在读者系统地了解整个软件开发过程之后，通过总结性回顾，进一步加深读者对软件工程的理解。

本书面向的主要读者对象包括从事软件开发的相关技术人员，以及学习“软件工程”课程的高等院校的软件工程专业低年级学生。建议读者在学习了《软件工程与计算（卷一）》后学习本书，将开发规模提升到小组开发级别的中等规模软件系统，重点展开对软件工程方法的学习。后续的《软件工程与计算（卷三）》进一步培养学生对软件工程方法和程序设计方法的实际运用能力，同时强化项目管理能力、团队交流沟通能力和对软件工程制品的整体把握能力。

骆斌老师主持策划了本书，参加了书稿写作的全部讨论，并对整个书稿的具体写作内容进行了指导和审阅。丁二玉老师主要负责编写了本书的第1章至第7章、第11章、第17~23章、附录A至附录D.2。刘钦老师主要负责编写了本书的第8章至第10章、第12章至第16章、附录D.3与附录D.4。最后丁二玉老师进行了全书的统稿工作。

前人工作是本书写作的基础，本书借鉴了已有著作和论文的内容，在此对列入引用文献清单的作者表示感谢。同时，本书在写作的过程中，得到了很多人士的帮助。感谢张瑾玉、黄蕾女士，郑滔、邵栋、任桐炜、刘嘉先生，他们参与了本书的内容讨论和评审，对本书提出了宝贵的意见和帮助。

限于编者的水平，错误和不妥之处在所难免，衷心希望读者指正赐教。如对本书有任何意见和建议，可通过 luobin@nju.edu.cn、eryuding@software.nju.edu.cn、qinliu@software.nju.edu.cn 与我们联系。

作者

2012年10月

南京大学北园

软件工程教材序

《软件工程与计算》使用说明

前言

第一部分 软件工程概论

第1章 软件工程基础	2
1.1 软件	2
1.1.1 软件独立于硬件	2
1.1.2 软件是一种工具	2
1.1.3 软件的核心是程序	3
1.1.4 软件开发远比编程要复杂	3
1.1.5 应用软件基于现实又高于现实	4
1.2 软件工程	4
1.2.1 定义	4
1.2.2 软件工程是一种工程活动	4
1.2.3 软件工程的动机	5
1.2.4 软件工程是科学性、实践性和工艺性并重的	6
1.2.5 软件工程追求足够好，不是最好	8
1.2.6 软件产品的产品是基于虚拟计算机的软件方案	8
1.2.7 软件工程的最终目的	9
1.3 软件工程概览	9
1.3.1 软件工程知识域	9
1.3.2 软件开发活动	11
1.3.3 软件工程的角色分工	13
1.4 习题	14

第2章 软件工程的发展	15
2.1 软件工程的发展脉络	15
2.2 20世纪50年代的软件工程	17
2.3 20世纪60年代的软件工程	19
2.4 20世纪70年代的软件工程	22
2.5 20世纪80年代的软件工程	26
2.6 20世纪90年代的软件工程	31
2.7 21世纪00年代的软件工程	37
2.8 习题	39

第二部分 项目启动

第3章 示例项目描述	42
3.1 背景	42
3.2 目标	42
3.3 系统用户	42
3.4 用户访谈要点	43
3.5 项目实践过程	47
第4章 项目管理基础	48
4.1 项目和项目管理	48
4.2 团队组织与管理	49
4.2.1 团队特征	49
4.2.2 团队结构	49
4.2.3 团队建设	51
4.3 软件质量保障	53
4.3.1 软件质量	53
4.3.2 质量保障	55

4.3.3	评审	56
4.3.4	质量度量	57
4.4	软件配置管理	57
4.4.1	配置管理动机	57
4.4.2	配置项	58
4.4.3	基线	58
4.4.4	配置管理活动	59
4.4.5	变更控制	60
4.5	项目实践	62
4.6	习题	62

第三部分 需求开发阶段

第5章	软件需求基础	66
5.1	引言	66
5.2	需求工程基础	67
5.2.1	需求工程简介	67
5.2.2	需求工程活动	67
5.2.3	需求获取	68
5.2.4	需求分析	69
5.2.5	需求规格说明	70
5.2.6	需求验证	70
5.2.7	需求管理	71
5.3	需求基础	71
5.3.1	需求	71
5.3.2	需求的层次性	71
5.3.3	结合层次性的需求开发	73
5.3.4	区分需求、问题域与规格说明	74
5.4	需求分类	75
5.4.1	需求谱系	75
5.4.2	软件需求的分类	76
5.5	项目实践	79
5.6	习题	80

第6章	需求分析方法	82
6.1	需求分析基础	82
6.1.1	需求分析的原因	82
6.1.2	需求分析模型	83
6.2	结构化分析	84
6.2.1	结构化分析方法	84
6.2.2	数据流图	85
6.2.3	实体关系图	90
6.3	面向对象分析	92
6.3.1	面向对象分析方法	92
6.3.2	用例	93
6.3.3	用例图	94
6.3.4	用例描述	96
6.3.5	概念类图(领域模型)	98
6.3.6	交互图(顺序图)	104
6.3.7	状态图	106
6.4	使用需求分析方法细化和明确需求	108
6.4.1	细化和明确需求内容	108
6.4.2	建立系统级需求	109
6.5	项目实践	112
6.6	习题	113
第7章	需求文档化与验证	115
7.1	文档化的原因	115
7.2	需求文档基础	115
7.2.1	需求文档的交流对象	115
7.2.2	用例文档	116
7.2.3	软件需求规格说明文档	116
7.3	需求文档化要点	118
7.3.1	技术文档写作要点	118
7.3.2	需求书写要点	119
7.3.3	软件需求规格说明文档书写要点	120
7.4	评审软件需求规格说明文档	121
7.4.1	需求验证与确认	121
7.4.2	评审需求的注意事项	121

7.5	以需求为基础开发系统测试用例	122
7.5.1	开发测试用例套件	123
7.5.2	开发测试用例	125
7.6	度量需求	125
7.7	将需求制品纳入配置管理	129
7.8	项目实践	130
7.9	习题	130

第四部分 软件设计

第8章	软件设计基础	133
8.1	软件设计思想的发展	133
8.2	软件设计的核心思想	134
8.3	理解软件设计	135
8.3.1	设计与软件设计	135
8.3.2	工程设计与艺术设计	136
8.3.3	理性主义和经验主义	137
8.3.4	软件设计的演化性	137
8.3.5	软件设计的决策性	138
8.3.6	软件设计的约束满足和多样性	138
8.4	软件设计的分层	139
8.5	软件设计过程的主要活动	140
8.6	软件设计的方法和模型	140
8.6.1	软件设计的方法	140
8.6.2	软件设计的模型	141
8.7	软件设计描述	141
8.7.1	设计视图和设计图	142
8.7.2	设计视角和设计关注	143
8.7.3	需求和涉众	144
8.7.4	设计理由	144
8.7.5	设计描述的模板	145
8.7.6	软件设计文档书写要点	145
8.8	项目实践	146
8.9	习题	146

第9章	软件体系结构基础	148
9.1	软件体系结构的发展	148
9.2	理解软件体系结构	148
9.2.1	定义	148
9.2.2	区分软件体系结构的抽象与实现	149
9.2.3	部件	150
9.2.4	连接件	150
9.2.5	配置	151
9.3	体系结构风格初步	153
9.3.1	主程序/子程序	153
9.3.2	面向对象式	154
9.3.3	分层	156
9.3.4	MVC	157
9.4	项目实践	159
9.5	习题	159
第10章	软件体系结构设计及构建	160
10.1	体系结构设计过程	160
10.1.1	分析关键需求和项目约束	160
10.1.2	选择体系结构风格	162
10.1.3	软件体系结构逻辑设计	162
10.1.4	软件体系结构实现	165
10.1.5	完善软件体系结构设计	170
10.1.6	定义构件接口	172
10.2	体系结构的原型构建	173
10.2.1	包的创建	173
10.2.2	重要文件的创建	174
10.2.3	定义构件之间的接口	174
10.2.4	关键需求的实现	175
10.3	体系结构集成与测试	175
10.3.1	集成的策略	175
10.3.2	桩、驱动与集成测试用例	177
10.4	软件体系结构设计文档描述	180
10.5	体系结构评审	181

10.6	项目实践	181	12.5	详细设计文档描述	212
10.7	习题	182	12.6	详细设计的评审	213
第11章 人机交互设计		183	12.7	项目实践	213
11.1	引言	183	12.8	习题	213
11.2	人机交互设计的目标	184	第13章 详细设计中的模块化与信息隐藏		215
11.3	人机交互设计的人类因素	185	13.1	模块化与信息隐藏思想	215
11.3.1	精神模型	185	13.1.1	设计质量	215
11.3.2	差异性	186	13.1.2	模块化与信息隐藏思想的 动机	215
11.4	人机交互设计的计算机因素	186	13.1.3	模块化与信息隐藏思想的 发展	216
11.4.1	可视化设计	186	13.2	模块化	217
11.4.2	常见界面类型	187	13.2.1	分解与模块化	217
11.5	人机交互设计的交互性	188	13.2.2	结构化设计中的耦合	219
11.5.1	导航	188	13.2.3	结构化设计中的内聚	220
11.5.2	反馈	189	13.2.4	回顾: MSCS 系统设计中的 模块化思想	221
11.5.3	一些人机交互设计原则	190	13.3	信息隐藏	222
11.6	人机交互设计过程	192	13.3.1	抽象与信息隐藏	222
11.6.1	基本过程	192	13.3.2	信息与隐藏	224
11.6.2	示例	193	13.3.3	模块说明	225
11.7	项目实践	195	13.3.4	回顾: MSCS 系统设计中的 信息思想	226
11.8	习题	195	13.4	习题	226
第12章 详细设计的基础		197	第14章 详细设计中面向对象方法 下的模块化		228
12.1	详细设计概述	197	14.1	面向对象中的模块	228
12.1.1	详细设计出发点	197	14.1.1	类	228
12.1.2	详细设计的上下文	198	14.1.2	类之间的联系	228
12.2	结构化设计	199	14.2	访问耦合	229
12.2.1	结构化设计的思想	199	14.2.1	访问耦合的分析	229
12.2.2	结构化设计的过程	200	14.2.2	降低访问耦合的方法	230
12.3	面向对象设计	202	14.3	继承耦合	234
12.3.1	面向对象设计的思想	202			
12.3.2	面向对象设计的过程	202			
12.3.3	通过职责建立静态模型	203			
12.3.4	通过协作建立动态模型	206			
12.4	为类间协作开发集成测试用例	210			

14.3.1	继承耦合的分析	234
14.3.2	降低继承耦合的方法	235
14.4	内聚	237
14.4.1	面向对象中的内聚	237
14.4.2	提高内聚的方法	238
14.5	耦合与内聚的度量	241
14.5.1	耦合的度量	241
14.5.2	内聚的度量	242
14.6	项目实践	243
14.7	习题	243

第15章 详细设计中面向对象方法 下的信息隐藏

15.1	封装类的职责	246
15.1.1	类的职责	246
15.1.2	封装——分离接口与实现	247
15.1.3	封装实现细节	247
15.2	为变更而设计	251
15.2.1	封装变更 / 开闭原则	251
15.2.2	多态	252
15.2.3	依赖倒置原则	253
15.2.4	总结	257
15.3	项目实践	257
15.4	习题	258

第16章 详细设计的设计模式

16.1	设计模式基础	262
16.2	可修改性及其基本实现机制	262
16.3	策略模式	265
16.3.1	典型问题	265
16.3.2	设计分析	266
16.3.3	解决方案	266
16.3.4	模式实例	268
16.4	抽象工厂模式	270
16.4.1	典型问题	270
16.4.2	设计分析	272

16.4.3	解决方案	272
16.4.4	模式实例	274
16.5	单件模式	276
16.5.1	典型问题	276
16.5.2	设计分析	276
16.5.3	解决方案	276
16.5.4	模式实例	277
16.6	迭代器模式	277
16.6.1	典型问题	277
16.6.2	设计分析	279
16.6.3	解决方案	279
16.6.4	模式实例	280
16.7	项目实践	280
16.8	习题	280

第五部分 软件构造、测试、 交付与维护

第17章	软件构造	282
17.1	概述	282
17.1.1	软件构造的定义	282
17.1.2	软件构造是设计的延续	283
17.2	软件构造活动	283
17.2.1	详细设计	283
17.2.2	编程	284
17.2.3	测试	284
17.2.4	调试	285
17.2.5	代码评审	286
17.2.6	集成与构建	288
17.2.7	构造管理	288
17.3	软件构造实践方法	288
17.3.1	重构	289
17.3.2	测试驱动开发	291
17.3.3	结对编程	295
17.4	项目实践	295

17.5 习题	296	19.2 测试层次	323
第18章 代码设计	298	19.2.1 测试层次的划分	323
18.1 设计易读的代码	298	19.2.2 单元测试	323
18.1.1 格式	298	19.2.3 集成测试	324
18.1.2 命名	302	19.2.4 系统测试	325
18.1.3 注释	302	19.3 测试技术	325
18.2 设计易维护的代码	304	19.3.1 测试用例的选择	325
18.2.1 小型任务	305	19.3.2 随机测试	326
18.2.2 复杂决策	306	19.3.3 基于规格的技术 ——黑盒测试方法	327
18.2.3 数据使用	308	19.3.4 基于代码的技术 ——白盒测试方法	329
18.2.4 明确依赖关系	308	19.3.5 特定测试技术	331
18.3 设计可靠的代码	308	19.4 测试活动	331
18.3.1 契约式设计	308	19.5 测试度量	334
18.3.2 防御式编程	310	19.6 项目实践	334
18.4 使用模型辅助设计复杂代码	310	19.7 习题	335
18.4.1 决策表	311	第20章 软件交付	336
18.4.2 伪代码	311	20.1 安装与部署	336
18.4.3 程序流程图	313	20.1.1 安装	336
18.5 为代码开发单元测试用例	313	20.1.2 部署	337
18.5.1 为方法开发测试用例	313	20.2 培训与文档支持	338
18.5.2 使用 Mock Object 测试类 方法	313	20.2.1 培训	338
18.5.3 为类开发测试用例	315	20.2.2 文档支持	339
18.6 代码复杂度度量	316	20.3 项目评价	341
18.7 问题代码	317	20.3.1 项目评价的原因	341
18.8 项目实践	318	20.3.2 项目评价的内容	341
18.9 习题	319	20.3.3 项目评价的方法	341
第19章 软件测试	320	20.3.4 注意事项	343
19.1 引言	320	20.4 项目实践	343
19.1.1 验证与确认	320	20.5 习题	344
19.1.2 软件测试的目标	321	第21章 软件维护与演化	345
19.1.3 测试用例	321	21.1 软件维护	345
19.1.4 桩与驱动	321	21.1.1 软件可修改性与软件维护	345
19.1.5 缺陷、错误与失败	322		

21.1.2	软件维护的类型	345	22.6	增量迭代模型	368
21.1.3	软件维护的高代价性	346	22.7	演化模型	370
21.1.4	开发可维护的软件	348	22.8	原型模型	371
21.1.5	软件维护过程与活动	349	22.9	螺旋模型	372
21.2	软件演化	350	22.10	Rational 统一过程	374
21.2.1	演化与维护	350	22.11	敏捷过程	376
21.2.2	软件演化定律	351	22.12	习题	378
21.2.3	软件演化生命周期模型与 演化活动	352	第23章 软件工程职业基础	379	
21.3	软件维护与演化的常见技术	354	23.1	软件工程职业	379
21.3.1	遗留软件	355	23.1.1	软件行业的发展	379
21.3.2	逆向工程	355	23.1.2	软件工程职业的出现	380
21.3.3	再工程	356	23.1.3	软件工程师职业素质	380
21.4	项目实践	356	23.2	软件工程职业概况	381
21.5	习题	357	23.2.1	知识体系	381
第六部分 软件过程模型 与职业基础					
第22章 软件开发过程模型	360	23.2.2	教育体系	381	
22.1	软件开发的典型阶段	360	23.2.3	职业道德规范	382
22.1.1	软件需求工程	360	23.2.4	认证体系	382
22.1.2	软件设计	361	23.2.5	行业协会	383
22.1.3	软件构造	361	23.3	软件工程的行业标准	383
22.1.4	软件测试	363	23.4	习题	384
22.1.5	软件交付	363	附录 A 软件需求规格说明文档模板	385	
22.1.6	软件维护	363	附录 B 文档注释规范	391	
22.2	软件生命周期模型	364	附录 C 软件工程道德和职业实践规范 (5.2版)的八项原则	395	
22.3	软件过程模型	364	附录 D 连锁商店管理系统 (MSCS) 相关文档	399	
22.4	构建-修复模型	366	参考文献	428	
22.5	瀑布模型	367			