

21世纪高等学校规划教材 | 软件工程

软件工程基础与实用教程

——基于架构与MVC模式的一体化开发

沈文轩 张春娜 曾子维 编著



清华大学出版社

21世纪高等学校规划教材 | 软件工程

软件工程基础与实用教程

——基于架构与MVC模式的一体化开发

沈文轩 张春娜 曾子维 编著



清华大学出版社
北京

内 容 简 介

本书是一本将软件工程基础理论与现代企业级开发技术相结合而诞生的软件工程实用教程。全书分为上下两篇。上篇是软件工程基础篇,系统地介绍了软件开发必须遵循的软件工程基本原理和开发方法,包括软件的需求、设计、实现、维护等开发过程和项目管理;下篇是体现本书特色的应用篇,介绍了应用企业级软件架构和 MVC 设计模式等现代开发技术进行软件开发过程,即基于架构与 MVC 模式的一体化开发方法。一体化开发方法是本书的核心内容,它一方面是指从分析、设计到实现全程遵循 MVC 模式,而不仅仅指某个开发阶段;另一方面是指架构与 MVC 模式的相互结合,架构可使得 MVC 的应用走向简单化、实用化。在一体化的基础上进而介绍了一种正交设计方法,它可以轻松处理复杂领域问题的解耦,将软件结构按 MVE 三大空间进行隔离设计,然后采用组装方式,以两大装配将 MVE 三大空间联系为一个软件整体。正交设计具有很强的规律性、可操作性。本书在最后介绍了一体化开发的应用案例,该案例不仅体现了一体化的具体应用,也展示了软件开发从抽象到细化的真实过程,让读者深入理解软件工程的一系列重要思想。

本书可以作为计算机或软件工程等相关专业的本科生、研究生的教材,同时也可作为软件开发领域的各类专业人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

软件工程基础与实用教程:基于架构与 MVC 模式的一体化开发/沈文轩等编著. —北京:清华大学出版社, 2012. 9

21 世纪高等学校规划教材·软件工程

ISBN 978-7-302-28733-9

I. ①软… II. ①沈… III. ①软件工程-高等学校-教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2012)第 089135 号

责任编辑:高买花 薛 阳

封面设计:傅瑞学

责任校对:李建庄

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京市清华园胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:29 字 数:722 千字

版 次:2012 年 9 月第 1 版 印 次:2012 年 9 月第 1 次印刷

印 数:1~3000

定 价:49.00 元

产品编号:046971-01

编审委员会成员

(按地区排序)

清华大学	周立柱 教授
	覃 征 教授
	王建民 教授
	冯建华 教授
	刘 强 副教授
北京大学	杨冬青 教授
	陈 钟 教授
	陈立军 副教授
北京航空航天大学	马殿富 教授
	吴超英 副教授
	姚淑珍 教授
中国人民大学	王 珊 教授
	孟小峰 教授
	陈 红 教授
北京师范大学	周明全 教授
北京交通大学	阮秋琦 教授
	赵 宏 教授
北京信息工程学院	孟庆昌 教授
北京科技大学	杨炳儒 教授
石油大学	陈 明 教授
天津大学	艾德才 教授
复旦大学	吴立德 教授
	吴百锋 教授
	杨卫东 副教授
同济大学	苗夺谦 教授
	徐 安 教授
华东理工大学	邵志清 教授
华东师范大学	杨宗源 教授
	应吉康 教授
上海大学	陆 铭 副教授
东华大学	乐嘉锦 教授
	孙 莉 副教授
浙江大学	吴朝晖 教授

扬州大学
南京大学

南京航空航天大学

南京理工大学
南京邮电学院
苏州大学

江苏大学
武汉大学
华中科技大学
中南财经政法大学
华中师范大学

江汉大学
国防科技大学
中南大学
湖南大学

西安交通大学

长安大学
哈尔滨工业大学
吉林大学

山东大学

中山大学
厦门大学
仰恩大学
云南大学
电子科技大学

成都理工大学

西南交通大学

李善平 教授
李 云 教授
骆 斌 教授
黄 强 副教授
黄志球 教授
秦小麟 教授
张功萱 教授
朱秀昌 教授
王宜怀 教授
陈建明 副教授
鲍可进 教授
何炎祥 教授
刘乐善 教授
刘腾红 教授
叶俊民 教授
郑世珏 教授
陈 利 教授
颜 彬 教授
赵克佳 教授
刘卫国 教授
林亚平 教授
邹北骥 教授
沈钧毅 教授
齐 勇 教授
巨永峰 教授
郭茂祖 教授
徐一平 教授
毕 强 教授
孟祥旭 教授
郝兴伟 教授
潘小轰 教授
冯少荣 教授
张思民 教授
刘惟一 教授
刘乃琦 教授
罗 蕾 教授
蔡 淮 教授
于 春 讲师
曾华乐 教授

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色

精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过二十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人: 魏江江

E-mail:weijj@tup.tsinghua.edu.cn

前言

这是一本基于软件工程原理、结合现代技术的软件开发实用教程。软件工程本是一门实践性非常强的学科，但是很多初学者在学习软件工程课程以后依旧感到茫然，面对实际项目不知所措。另外，对于一些现代技术的学习和应用没有上升到软件工程的高度去理解和认识。本书以掌握软件工程基本原理为指导应用，应用要与现代技术相结合的目标来讲解软件工程，全书分为上、下两篇：上篇为基础篇，系统地介绍了软件开发必须遵循的软件工程基本原理和开发方法；下篇为应用篇，首先对企业级软件架构和 MVC 设计模式等现代开发技术进行介绍，然后讲述由此带来的开发方法上的变化及在软件工程中的使用。

近几年，诸如微软的.NET、Sun 的 J2EE 等软件架构技术在企业级软件开发中得到广泛应用，使得软件开发效率、质量、维护性、成功率得到空前的提高，其商业价值得到 IT 业界的认可，以至在业界形成了汹涌的应用浪潮。

软件行业发展的一个特殊规律是技术催生方法学，就是先使用技术后讲究方法。如 20 世纪 60 年代，出现了 FORTRAN、COBOL、Lisp 等过程性高级语言^[1]，直到 20 世纪 70 年代，随着 BASIC、Pascal、C 的广泛应用，诞生了结构化的程序设计方法学。到 20 世纪 80 年代 C++ 已经广泛应用，而到 20 世纪 90 年代，面向对象的开发方法 OOA 和 OOD 才趋于普及和成熟。回顾以往，方法学的诞生比技术的兴起延时 5~10 年。架构技术与基于架构的开发方法也面临这样的窘境，一方面企业开发在大量应用架构，另一方面它的开发方法还没有规范。目前在技术层面推进上，有关架构的技术书籍随处可见，这些书籍和作者在技术推广方面功不可没；高等教育的课堂教学，以及适应就业需求“揭竿而起”的就业培训也是把“热点”集中在架构的编程技术层面。然而，在基于架构技术的方法学层面上的教学和教材还不多见。在很多软件工程的经典教材中，甚至在面向 21 世纪的新教材中还是停留在传统的面向对象开发方法上，主要讲述经典的 Booch 方法、Coad-Yourdon 方法、Ivar Jacobson 的 OOSE 方法、James Rumbaugh 的 OMT^{[2][3]}。有些先进的教材中介绍了统一方法，但也不是基于架构的，面对今天的架构它们显得能力不足。试想一下，如果基于架构的技术却采用 OMT 方法开发，结果是把简单的事情复杂化，适得其反，架构所带来的降低开发难度的好处将荡然无存。

本书参考了许多开发团队的成功经验，结合作者在应用架构方面的研究和工程实践，在方法学的层面，介绍有关架构与 MVC 模式统一起来在软件的分析、设计、实现的开发过程中具体的应用方法和步骤，也就是本书副标题“基于架构与 MVC 模式的一体化开发”。为完整体现一体化开发方法的应用，在第 13 章介绍了一个教师和学生都熟悉的“毕业设计流程管理系统”案例。尽管该案例采用 J2EE 架构，但它同样适合其他架构。

一体化开发的核心思想是：从系统分析开始就在按 MVC（更准确地说是 MVE）划分，直到设计、实现和维护的全过程乃至开发团队的组建都遵循 MVC 的理念（分离+协作）。一体化开发的目的是：先进的开发方法要与先进的技术相适应，降低开发难度，解

放生产力。

上篇共有 10 章，本书一改传统的讲述方式，在讲述内容与章节上做了重大调整。由于结构化与面向对象的分析和设计遵循的原理、基础是相同的，把软件分析和软件设计基础部分单独抽取出来，然后分别从结构化、面向对象两个层次讲述从分析到设计。读者要重点理解和掌握第 7 章的“面向对象从分析到设计”。基础篇也包括软件测试和维护的内容，第 10 章介绍了软件项目管理。

下篇是本书具有特色的应用篇，也是体现本书从实用出发的核心内容：应用架构与 MVC 设计模式相结合的一体化开发方法。

第 11 章就是作为一体化开发的基础篇，对架构与 MVC 模式从基本概念开始直到它们如何达到统一做了系统的介绍。比如读者想追根寻源地了解这样的系列问题：什么是架构？企业为什么采用架构？架构给软件开发提供了什么？如何使用？架构为什么分层？分为多少层？每层起什么作用？……什么是设计模式？什么是 MVC 设计模式？MVC 分层与架构分层有何区别和联系？等等。理解这些概念将有助于加深对架构、设计模式方面的理解。为了让初学者容易理解，本书尽量避免使用晦涩的学术性用语，而是采用通俗、形象的语言来解释概念，引用生活中熟悉的例子加以印证。对于从其他学习中已建立了一些基本概念、有一定基础的读者，阅读本部分内容来系统梳理一下思路也将大有裨益。11.3 节具体地介绍了一个常用的企业架构 J2EE；另外，为开阔读者眼界，在 11.4 节对企业集成架构 SOA、SCA 进行了介绍，在 11.7 节介绍了开源框架 SSH 在 MVC 模式中的使用。11.6 节是第 11 章的核心内容，在以架构与 MVC 的基本概念为铺垫下，重点从原理上叙述了架构分层与 MVC 模式分层的区别和联系，找到了它们之间的共同点，也就是演奏了基于架构与 MVC 模式一体化开发的前奏曲。此外，为了帮助读者更好地理解 MVC 的特征，也对事件驱动模式和观察者模式做了一定的介绍。

第 12 章是本书的核心内容，介绍一体化开发过程的基本原理、具体实施。第 12 章实际就是软件开发从面向过程到面向对象再升华达到的第三个层次：面向架构从分析到设计。一体化一方面指架构的分层与 MVC 分层的统一，架构与 MVC 设计模式是天作之合，架构解决体系结构设计难题，MVC 解决领域结构设计的困扰；另一方面指从需求分析、概要设计、详细设计到编码实现全过程始终遵循 MVC 的思想。这种一体化方法使得软件开发从对客观抽象到具体实现的逐步精化过程具有很强的简便性、规律性、可操作性。它对于软件工程初学者不再虚无缥缈，对于软件分析师、设计师乃至程序员在实际工作中都具有很好的参考价值。在领域问题软件结构设计方面，结合 MVC 的事件响应管理模型，介绍了正交设计方法。该方法首先在水平方向按软件体系结构层找出每个事件的层间的依赖关系即执行链，然后按垂直方向分为 V (Viewer)、M (Model)、E (Entity) 三大结构进行分层设计。这种正交设计方法克服了传统的对象识别和对象设计上的困难。在界面和事件的命令组织设计方面也采用了正交设计方法，使得繁杂的设计工作变得井然有序。

第 13 章是一体化开发方法的应用案例。案例采用 Java EE 架构，读者可以完整体验基于架构的从分析到设计的过程遵循一体化开发方法。为节省篇幅，案例中对于内容不同但如法炮制的部分做了大幅削减。

参考本书提供的一体化方法和实际案例，读者将从技术层面提升到方法学层面来理解和使用架构；帮助读者领悟到架构给方法学所带来的进步；通过与读者分享案例来学习这

种软件开发方法，体验由难变易、由繁变简、逐步精化、有章可循，走向简单化、条理化、规范化的可贵提升。

为了解架构与 MVC 的关系，本书介绍了很多有关架构方面的知识，同时也介绍了常用的架构 J2EE 和 SSH 框架，并以代码的形式介绍了相关例子。但本书是以理解 MVC 模型为出发点，而不是系统性地讲解编程技术和技巧。如果阅读代码部分有困难，或者想从具体编程上有待提高的读者还要参阅其他相关资料。

很多读者在学习软件工程课程之后，感到空洞茫然，面对实际问题无所适从；也有些读者在应用具体的技术之中缺乏软件工程的思想，对技术的理解陷入片面，技术发挥受到局限。本书意在学习软件工程和架构技术之间搭建一个桥梁。本来软件工程和架构技术之间存在着天然的联系，但是由于课程设置、教学目标或因教学对象不同，造成两者知识的断裂。因此，本书对单独学习软件工程或单独学习架构编程各自存在的不足进行了弥合，可以作为这两门知识的双向理解，最终将两门知识融合在一起，提升对软件开发的理解和解决问题的能力。本书可以作为本科的专业课教材，同时也是毕业设计极好的参考资料；本书也适合研究生深入学习和研究架构的基础教材；本书对那些在工作岗位上需要进一步理解架构、理解开发方法的从业者也有很好的帮助。

尽管本书在编著过程中力求内容精炼、语言准确，但由于时间仓促和作者水平所限，疏漏与不足之处在所难免，欢迎广大读者批评指正。

马云吉、王莉、孙学波、张文宇、李迎春、郭秋、张秀梅、张晓霞、芦明、王博等老师参加了本书的编写和校对工作，在此表示感谢。

作者

2012年7月于辽宁科技大学



目 录

上篇 软件工程基础篇

第 1 章 软件工程概述	3
1.1 软件的概念	3
1.1.1 软件和软件产品	3
1.1.2 软件产品的类型	4
1.2 软件的发展与软件危机的产生	6
1.2.1 软件的发展过程	7
1.2.2 软件危机	8
1.2.3 软件危机的表现	9
1.3 解决软件危机的途径	9
1.4 软件工程的定义	10
1.5 软件工程的内容	11
1.5.1 从计算机科学视角看软件工程	11
1.5.2 从工程视角看软件工程	12
1.5.3 从管理视角看软件工程	13
1.5.4 从基本原理视角看软件工程	14
1.6 软件发展的新阶段与新问题	15
1.6.1 软件发展的第 4 阶段和第 5 阶段	15
1.6.2 以面向对象为基础的面向架构技术	16
1.6.3 以业务单元为基础的面向服务架构 SOA	16
1.6.4 以软件作为服务的应用模式	17
1.7 软件工程的演变发展	19
1.7.1 从软件技术到软件工程	19
1.7.2 从传统软件工程到现代软件工程	20
1.7.3 从软件工程学到软件经济学	23
1.7.4 从软件工程应用到教学	24
本章小结	24
习题	25
第 2 章 软件过程	27
2.1 软件过程概述	27

2.1.1	软件过程定义	27
2.1.2	软件过程的活动集合	27
2.1.3	软件过程贯穿软件工程方法论	28
2.2	软件过程与软件生存周期	29
2.2.1	软件生存周期	29
2.2.2	软件生存周期与软件过程的关系	30
2.3	软件过程与软件工程	30
2.3.1	工程是管理过程的	30
2.3.2	软件过程是软件工程的子集	30
2.4	软件开发模型	31
2.4.1	瀑布模型	31
2.4.2	快速原型模型	32
2.4.3	增量模型	33
2.4.4	螺旋模型	34
2.4.5	统一过程 RUP	35
2.4.6	敏捷开发与极限编程	40
2.4.7	关于 RUP 与 XP 之争	44
2.5	软件过程与开发方法	44
2.5.1	软件开发的本质与开发方法	44
2.5.2	面向过程开发方法	45
2.5.3	面向对象开发方法	45
2.5.4	面向组件开发方法	45
2.5.5	面向服务开发方法	46
2.5.6	开发方法进步带来的思考	46
2.6	软件过程改进与 CMM	47
2.6.1	过程改进 SPI	47
2.6.2	软件能力成熟度模型 SW-CMM	48
	本章小结	51
	习题	51
第 3 章	软件计划	53
3.1	软件计划概述	53
3.2	问题定义	54
3.3	可行性分析	56
3.3.1	可行性分析概要介绍	56
3.3.2	可行性分析的方法	57
3.3.3	可行性分析工具之 SFD	60
3.3.4	可行性分析之业务建模	62

3.3.5 可行性分析报告	67
3.3.6 软件项目计划	68
本章小结	68
习题	69
第 4 章 软件需求基础	71
4.1 需求分析概述	71
4.1.1 需求分析的必要性	71
4.1.2 需求分析的内容和步骤	73
4.2 需求获取	75
4.2.1 需求获取的内容和手段	75
4.2.2 需求的层次	75
4.2.3 需求获取方法与策略	76
4.3 需求建模	78
4.3.1 需求建模的意义	78
4.3.2 需求建模的过程	79
4.3.3 需求建模的方法	79
4.4 需求说明	81
4.4.1 需求说明的作用	81
4.4.2 需求说明质量要求	81
4.4.3 需求规格说明的编制	83
4.5 需求评审	85
4.5.1 需求评审概述	85
4.5.2 需求评审举例说明	87
4.6 需求变更管理	88
4.6.1 需求变更与需求管理	88
4.6.2 变更管理组织与流程	89
本章小结	90
习题	90
第 5 章 软件设计基础	91
5.1 软件开发的逐步求精	91
5.1.1 逐步求精的概念	91
5.1.2 软件开发的求精过程	92
5.1.3 过程求精与数据求精	92
5.2 软件设计过程和任务	94
5.2.1 软件设计的输入和输出	94
5.2.2 软件设计的总体任务	94

5.2.3 软件设计的文档	95
5.3 模块化与信息隐藏	96
5.3.1 模块和模块化	96
5.3.2 模块分解原理	96
5.3.3 信息隐藏思想	97
5.3.4 模块独立性度量	97
5.3.5 模块优化的一些概念	100
5.4 详细设计	101
5.4.1 详细设计的作用	101
5.4.2 详细设计的内容	102
5.4.3 详细设计的表达方式	103
5.4.4 详细设计说明书	105
本章小结	105
习题	105
第 6 章 面向结构化从分析到设计	107
6.1 结构化开发概述	107
6.2 面向数据流的结构化分析	108
6.2.1 数据流图	109
6.2.2 DFD 的分析思想	110
6.2.3 DFD 举例	111
6.2.4 数据字典	112
6.3 面向数据流的结构化设计	113
6.3.1 从需求模型到设计模型	113
6.3.2 软件结构图	114
6.3.3 从 DFD 映射到 SC	115
本章小结	117
习题	117
第 7 章 面向对象从分析到设计	118
7.1 面向对象开发简介	118
7.1.1 面向对象的基本概念	118
7.1.2 面向对象的软件开发	122
7.1.3 面向对象开发的优点	124
7.1.4 面向对象是走向高端的基础	126
7.2 统一建模语言 UML	126
7.2.1 建模语言的概念	126
7.2.2 UML 发展史	127

7.2.3	UML 图形符号概览	129
7.2.4	用例图	129
7.2.5	类图与对象图	131
7.2.6	包图	142
7.2.7	活动图与状态图	146
7.2.8	对象交互图	150
7.2.9	组件图与部署图	154
7.2.10	UML 视图简介	157
7.3	面向对象分析 (OOA)	159
7.3.1	OOA 方法的统一	159
7.3.2	统一方法的 OOA 模型	161
7.3.3	用例模型	161
7.3.4	概念模型	166
7.4	面向对象设计	170
7.4.1	OOD 概述	170
7.4.2	OOD 模型	172
7.4.3	问题域部分的设计	173
7.4.4	界面交互设计	180
7.4.5	数据管理部件设计	186
7.4.6	任务管理设计	194
	本章小结	196
	习题	197
第 8 章	软件实现	199
8.1	编码	199
8.1.1	编程语言	199
8.1.2	编码风格	200
8.1.3	调试	200
8.2	测试	201
8.2.1	测试的基本概念	201
8.2.2	测试计划	205
8.2.3	测试方法	207
8.2.4	测试过程	217
8.2.5	测试设计与测试报告	224
8.2.6	测试工具	227
	本章小结	229
	习题	230

第 9 章 软件维护	232
9.1 软件维护概述.....	232
9.1.1 软件维护的概念.....	232
9.1.2 软件维护的重要性.....	232
9.2 软件维护的分类.....	233
9.2.1 纠错性维护.....	233
9.2.2 适应性维护.....	233
9.2.3 改善性维护.....	233
9.2.4 预防性维护.....	234
9.2.5 各种维护的比重.....	234
9.3 软件的可维护性.....	234
9.3.1 维护的副作用及困难.....	234
9.3.2 软件结构化维护.....	236
9.4 软件维护的实施过程.....	237
9.4.1 维护组织和流程.....	237
9.4.2 维护的决策流程.....	238
9.4.3 维护的报告与审核.....	240
9.4.4 保存维护记录.....	240
9.4.5 评价维护活动.....	241
9.4.6 可维护性复审.....	241
9.5 配置管理.....	241
9.5.1 软件配置.....	242
9.5.2 软件配置管理.....	242
9.5.3 配置管理的功能.....	244
9.5.4 配置管理工具.....	245
本章小结.....	247
习题.....	247
第 10 章 软件项目管理	248
10.1 项目和项目管理的基本概念.....	248
10.1.1 项目的意义.....	248
10.1.2 软件项目管理过程.....	250
10.2 软件度量.....	252
10.2.1 软件度量概述.....	252
10.2.2 面向代码行的度量.....	255
10.2.3 面向功能的度量.....	256
10.3 软件估算.....	261

10.3.1	软件估算概述	261
10.3.2	软件规模估算	264
10.3.3	软件成本估算	266
10.4	开发进度计划	271
10.4.1	项目计划与进度计划	271
10.4.2	时间管理与进度计划	272
10.4.3	进度计划的安排	273
10.4.4	开发进度管理	285
10.5	团队组织与人力资源计划	286
10.5.1	任务分解与人员配备	286
10.5.2	人员配备规则	289
10.5.3	项目组织	291
10.5.4	项目经理	293
	本章小结	294
	习题	295

下篇 软件工程实用篇

第 11 章	架构与 MVC 模式	299
11.1	软件架构	299
11.1.1	软件架构与企业软件架构	300
11.1.2	企业架构解决软件开发的什么问题	302
11.1.3	架构的容器与组件	305
11.1.4	流行的企业架构简介	307
11.2	架构的分层	308
11.2.1	架构为什么分层	309
11.2.2	基本的三层结构	309
11.2.3	客户层与业务层	310
11.2.4	数据层与 DAO	311
11.2.5	业务层与 DAO 层	315
11.2.6	N 层体系结构	316
11.3	Java EE 架构简介	317
11.3.1	J2EE 架构概览	317
11.3.2	J2EE 容器和组件	318
11.3.3	J2EE 分层模型	323
11.3.4	J2EE 的发展	324
11.3.5	J2EE 的优势	326
11.4	面向企业集成的架构简介	327