

面向计算机科学与技术专业规范系列教材



软件工程概论

郑人杰 马素霞 殷人昆 编著



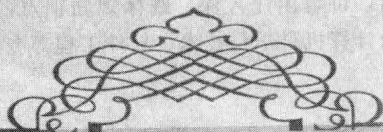
3
*An Introduction to
Software Engineering*



机械工业出版社
China Machine Press

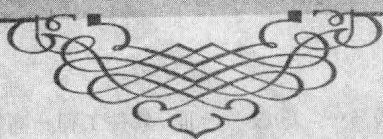


面向计算机科学与技术专业规范系列教材



软件工程概论

郑人杰 马素霞 殷人昆 编著



*An Introduction to
Software Engineering*



机械工业出版社
China Machine Press

软件工程学科具有知识面广、发展迅速、实践性强等特点。本书作者针对软件工程的学科特点，在系统讲解软件工程理论、方法和工具的同时，注重结合实例分析软件工程方法、技术和工具的综合应用；在兼顾传统的结构化方法的同时，注重介绍广为采用的面向对象方法。全书内容组织成四部分：第一部分是软件工程概述，第二部分介绍传统的软件开发方法，第三部分讲述面向对象的软件开发方法，第四部分介绍软件维护与软件管理。

本书结构合理、内容丰富，讲解由浅入深，既体现知识点的连贯性、完整性，又体现知识在实际中的应用，适合作为计算机科学与技术、软件工程等专业本科生的教材。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目 (CIP) 数据

软件工程概论 / 郑人杰, 马素霞, 殷人昆编著. —北京: 机械工业出版社, 2009. 11
(面向计算机科学与技术专业规范系列教材)

ISBN 978-7-111-28381-2

I. 软… II. ①郑… ②马… ③殷… 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 (2009) 第 172872 号

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: 刘立卿

北京市荣盛彩色印刷有限公司印刷

2010 年 3 月第 1 版第 1 次印刷

185mm×260mm·23.25 印张

标准书号: ISBN 978-7-111-28381-2

定价: 36.00 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

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzjsj@hzbook.com

出版者的话

机械工业出版社华章公司是国内重要的教育出版公司，培生教育集团（拥有 Addison Wesley、Prentice Hall 等品牌）是全球知名的教育出版集团，双方在过去长达十余年的合作中秉承“全球采集内容，服务教育事业”的理念，遴选、移译了国外大量的在计算机科学界享誉盛名的专家名著与名校教材，其中包括 Donald E. Knuth、Alfred V. Aho、Jeffrey D. Ullman、John E. Hopcroft、Dennis Ritchie 等大师名家的经典作品（收录在大理石封面的“计算机科学丛书”中），这些作品对国内计算机教育及科研事业的发展起到了积极的促进作用。

随着国内计算机科学与技术专业学科建设的不断完善、教学研究的蓬勃发展，以及教材改革的逐渐深化，计算机科学与技术专业的优秀课程及教材不仅仅是“引进来”（版权引进），而且需要“走出去”（版权输出）了。

近几年以来，教育部计算机科学与技术专业教学指导分委员会根据我国计算机专业教育的现状以及社会对人才的需求，发布了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》（以下简称《规范》）。为配合《规范》的实施推广，同时为落实中央“提高高等教育质量”的最新指导思想，在教育部计算机科学与技术专业教学指导分委员会的指导下，在国内知名高校众多教授的帮助下，我们出版了这套“面向计算机科学与技术专业规范系列教材”。

本套教材的作者在长达数十年的科研和教学经历中积累了丰富的知识和经验，也奠定了他们在学术和教学领域的地位，教材的内容体现了他们的教学思想和教学理念，本套教材也是传承他们优秀教学成果的最好载体，是中国版的专家名著和名校教材，相信它们的出版对提高计算机科学与技术专业的教育水平和教学质量能够起到积极的作用。

华章与培生作为专业的出版团队，愿与高等院校的老师共同携手，在这套教材的出版上引进国际先进教材出版经验，在教学配套资源的建设上做出新的尝试，为促进中国计算机科学与技术专业教育事业的发展，为增进中国与世界文化的交流而努力。





序 言

近 20 年里，计算机学科有了很大的发展，人们普遍认为，“计算机科学”这个名字已经难以涵盖该学科的内容，因此，改称其为计算学科（Computing Discipline）。在我国本科教育中，1996 年以前曾经有计算机软件专业和计算机及应用专业，之后被合并为计算机科学与技术专业。2004 年以来，教育部计算机科学与技术专业教学指导分委员会根据我国计算机专业教育和计算学科的现状，为更好地满足社会对计算机专业人才的需求，发布了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》（以下简称《规范》），提出在计算机科学与技术专业名称之下，构建计算机科学、计算机工程、软件工程和信息技术四大专业方向。《规范》中四大专业方向的分类，在于鼓励办学单位根据自己的情况设定不同的培养方案，以培养更具针对性和特色的计算机专业人才。

为配合《规范》的实施，落实中央“提高高等教育质量”的精神，我们规划了“面向计算机科学与技术专业规范系列教材”。本系列教材面向全新的计算学科，针对我国高等院校逐步向新的计算机科学与技术专业课程体系过渡的趋势编写，在知识选择、内容组织和教学方法等方面满足《规范》的要求，并与国际接轨。本套教材具有以下几个特点：

（1）**体现《规范》的基本思想，满足其课程要求。**为使教材符合我国高等院校的教学实际，编委会根据《规范》的要求规划本套教材，广泛征集在国内知名高校中从事一线教学和科研工作、经验丰富的优秀教师承担编写任务。

（2）**围绕“提高教育质量”的宗旨开发教材。**为了确保“精品”，本系列教材的出版不走盲目扩大的路子，每本教材的选题都将由编委会集体论证，并由一名编委担任责任编委，最大程度地保证这套教材的编写水准和出版质量。

（3）**教材内容的组织科学、合理，体系得当。**本套教材的编写注重研究学科的新发展和新成果，能够根据不同类型人才培养需求，合理地进行内容取舍、组织和叙述，还精心设计了配套的实验体系和练习体系。

（4）**教材风格鲜明。**本套教材按 4 个专业方向统一规划，分批组织，陆续出版。教材的编写体现了现代教育理念，探讨先进的教学方法。

（5）**开展教材立体化建设。**根据需要配合主教材的建设适时开发实验教材、教师参考书、学生参考书、电子参考资料等教辅资源，为教学实现多方位服务。

我们衷心希望本系列教材能够为我国高等院校计算机科学与技术等专业的教学作出贡献，欢迎广大读者广为选用。

“面向计算机科学与技术专业规范系列教材”编委会

面向计算机科学与技术专业规范系列教材

编委会

主任委员： 蒋宗礼

副主任委员： 王志英 钱乐秋

委员：（以姓氏拼音为序）

陈道蓄	陈 明
傅育熙	何炎祥
黄刘生	贾云得
姜守旭	李仁发
李晓明	刘 辰
马殿富	齐 勇
孙吉贵	孙茂松
吴功宜	吴 跃
谢长生	于 戈
张 钢	周兴社

秘书组： 温莉芳 刘立卿 姚 蕾

本书责任编辑： 钱乐秋



前言

当今，软件业是社会经济发展的先导性和战略性产业，它已成为信息产业和国民经济新的增长点和重要支柱。软件工程在软件开发中起着重要的作用，对软件产业的形成及发展起着决定性的推动作用。采用先进的工程化方法进行软件开发和生产是实现软件产业化的关键技术手段。与其他产业相比，软件产业具有自己的特殊性。软件产业的发展更加依赖于人力资源，因此软件产业的竞争越来越集中到对人才的竞争。然而，刚毕业的大学生往往要经过半年到一年的培训才能适应软件企业的工作。长期以来，我国软件人才的现状远远不能满足软件产业发展的要求。因此，软件工程人员队伍的成长，特别是高层软件工程人员队伍的成长显得更为紧迫。

自从软件工程概念诞生以来，学术界和工业界做了大量的研究与实践工作，也取得了许多重要成果。尤其是上个世纪 90 年代以后，随着网络技术及面向对象技术的广泛应用，软件工程取得了突飞猛进的发展。软件工程已从计算机科学与技术中脱离出来，逐渐形成了一门独立的学科。软件工程教育所处的地位也越来越重要。软件工程课程已成为软件工程、计算机科学与技术等专业的必修课程。

软件工程课程是实践性比较强的课程，如果学生没有实践经验，则很难理解相关的理论知识。因此，教师普遍感到软件工程课程难教，而学生则普遍感到难学。近年来，软件工程学科的发展非常迅速，新的理论、方法和工具层出不穷，其中很多已经应用到企业的实际工作中。软件工程的的教学面临越来越大的压力。我们认为，除了需要在教学内容、教学方法方面进行改革之外，实践能力的培养对于建设一支企业需要的合格软件工程人才队伍显得更为关键。

我们在编写中力图遵循如下原则：

(1) 既要强调和突出基本概念、基本方法，又要尽可能使材料内容的组织符合学生的认识规律，在讲解理论的过程中尽量结合实例，并注重软件工程方法、技术和工具的综合应用，避免讲解抽象和枯燥。

(2) 在兼顾传统的结构化方法的同时，注重当前广为采用的面向对象方法。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) 既要充分重视技术性内容，使其作为初学者必须掌握的知识，同时也要兼顾软件工程实践中必不可少的管理知识。

总之，本书力争做到结构合理、内容丰富，讲解由浅入深，既体现知识点的连贯性、完整性，又体现其在实际中的应用。

作者

2009 年 12 月

第1章 软件与软件工程的概 念 (3~4 学时)

该章主要是概念性和基础性的内容,包括软件的概念、特性及软件危机的主要表现,软件工程的概
念、软件生存期及典型的生存期模型等。这些内容应做重点讲解。

最后一节是软件工程知识体系及知识域的介绍,属于了解性的内容,可以让学生课后
阅读相关的参考文献。

第2章 软件工程方法与工具 (2~3 学时)

该章对主要的软件工程方法进行了介绍,包括传统的结构化方法、面向对象方法及形
式化方法。并对面向对象的概念与开发方法进行了介绍,这部分内容在面向对象程序设计
课程中基本都已涉及,这里属于复习。

对 UML 的介绍也只是导引性的,详细的介绍在第 7 章讲解。鉴于软件工具是软件工
程的三个要素之一,而且在软件开发中起着越来越重要的作用,因此本书对软件工具进行
了介绍,但由于涉及的工具很多,无法一一进行介绍,本书重点介绍了业界广泛使用的
Rational Rose UML,建议学生能够在课后进行软件安装,并熟悉软件的基本使用。

第3章 软件需求获取与结构化分析方法 (4~6 学时)

3.1 节和 3.2 节需要详细讲解,也需要学生较好地准确掌握。3.1 节的内容不仅适用
于传统方法,也适用于面向对象的方法。3.2 节对数据流分析方法做了较详细的介绍,画
分层的数据流图、E-R 图及状态图是本节的重点,书中结合实例进行了讲解。学生在课后
应该实践才能逐渐掌握上述知识,在实践中最容易出现的问题是数据流的分层,由抽象到
具体的程度难以把握,很多同学一开始就画出非常详细的数据流图,往往图太复杂,看不
清楚,失去了建模的意义。如果层次难以把握,最起码应强调先画出顶层数据流图,正确
的顶层数据流图是一个好的开端,也是设计阶段中外部接口设计和交互设计的依据。

第4章 结构化设计方法 (5~6 学时)

在传统的方法中,体系结构设计实质上就是软件结构设计或模块结构设计。该章 4.3
节讲解了基于数据流的设计方法,重点讲解了使用变换型映射方法和事务型映射方法生成
初始的模块结构,以及如何对模块结构进行改进,最后给出了一个设计与改进的例子。这
部分内容需要重点讲解。

对于大多数应用系统来说,人机交互(用户)界面设计是接口设计的主要内容。人机
交互界面的设计越来越受到重视,因为很多使用方面的特性都体现在用户界面上,用户往
往会根据界面来评价软件。这方面是最难于把握的,因为不同的人会有不同的审美观点和
不同的使用习惯。4.4 节介绍了对界面类型及人机交互的准则。

4.5 节介绍了文件设计和数据库设计。在数据库设计部分主要是讲解如何将 E-R 图映

射到数据库中的表。考虑到数据库的具体设计在数据库原理与应用等课程中学过，因此本节没有再展开。

4.6 节是需要重点讲解和实践的内容。老师可能会认为算法的描述在程序设计类课程、数据结构与算法课程中都学习和实践过，在软件工程课程中不需要重点介绍。但在实际项目开发中，大多数学生在这方面都表现不佳。算法图中的错误很多，很多情况下都不是结构化的，往往要经过多次修改才能符合要求。自顶向下、逐步细化的方法也运用得不好，致使在描述复杂算法时很困难。

第5章 编码 (3~4 学时)

该章重点讲解编码风格和编码规范。大多数初学者和软件开发新手喜欢学习编码技术，但不愿意花时间写注释，很多方面更喜欢随心所欲，不愿意受规范的束缚。主要原因是他们并没有认识到风格和规范的重要性，认为实现了功能就万事大吉了。对于学生完成作业和练习来说，可以是这样认为，因为别人不需要看，以后也不需要再修改了。但对于企业开发的系统来说，完成了功能才只是开始，以后面临的是漫长和繁重的维护任务。况且现代软件开发都是团队开发，如何使很多人开发的代码如同一个人开发的一样，使得互相都能读懂、易于阅读，解决的办法就是都要遵循统一的规范。

第6章 软件测试方法 (6 学时)

该章需要学生了解与软件测试相关的概念、软件测试的重要性、软件测试与开发各个阶段的关系，同时也要了解穷举测试是不可能的，测试只能证明软件存在错误，而不能证明软件中不存在错误。白盒测试技术及黑盒测试技术是需要重点讲解的内容，也是读者必须掌握的内容。

第7章 统一建模语言 UML 概述 (4 学时)

该章对 UML 的基本模式、事物、关系及建模时用到的各种图进行了介绍。由于采用面向对象方法建模普遍使用 UML 来描述，因此该章需要重点讲解。后面在面向对象分析、面向对象设计中虽然会涉及，但在那里基本上都是使用，细节的地方没有再讲解。

在组织教学时教师可以根据需要对顺序进行调整，将有些部分移到后面的章节中，与分析 and 设计结合起来讲解。

第8章 面向对象分析 (4 学时)

该章主要讲解面向对象分析的 3 种模型，即功能模型、静态模型和动态模型（交互模型）。在讲解中结合了实例，并将面向对象的分析方法与 Rational Rose 工具的使用有机地结合起来，避免了理论讲解太抽象和枯燥。

该章的内容需要重点讲解，学生需要在课后进行实践。

第9章 软件体系结构与设计模式 (4~6 学时)

该内容比较抽象，有一定难度，大多数本科生接受起来比较困难，教师可以根据具体情况有选择地进行讲解，如认为不必要或学时有限也可以不讲。

第10章 面向对象设计 (4~6 学时)

面向对象设计是面向对象分析的继续，也是面向对象分析的深入和细化，重点是对面向对象分析时建立的对象模型进行调整和细化，另外需要考虑与实现有关的方面，如数据存储设计、人机交互界面设计等。

从总体上来说，与传统的结构化方法相比，面向对象设计与面向对象分析具有更好的连续性，在方法和工具的使用上保持了高度的一致。限于篇幅，该章没有再讲 Rational Rose 的使用。但学生在课后应该将其结合起来进行实践。

第 11 章 软件维护 (2~4 学时)

该章对软件维护的任务、软件维护活动及软件维护方法进行了介绍。要求学生了解为什么要对软件进行维护，有哪些种类的维护以及软件维护的重要性。维护不仅仅是修改程序，也需要遵循一定的过程。该章内容也可由教师根据教学安排的实际情况做裁剪和取舍。

第 12 章 软件过程与软件过程改进 (2~3 学时)

该章介绍了软件过程的定义、分类，软件过程改进模型 CMM/CMMI 以及个人软件过程 (PSP) 和团队软件过程 (TSP)。该章属于了解性内容，如果学时有限，可以有选择地进行介绍。

第 13 章 软件项目管理 (4~6 学时)

该章重点讲解软件生产率和质量的度量、软件项目的时间管理、软件项目的成本管理以及软件配置管理。人员管理和风险管理部分可以让学生课后阅读。

第 14 章 软件工程标准及软件文档 (2 学时)

软件工程的标准化工作对于推动软件的产业化具有重要意义，而文档是软件不可分割的一部分，对软件的维护起着重要作用。该章要求学生了解软件工程标准的分类和意义，从而增强软件工程项目的标准化意识，为今后工作中的应用打下基础。此外，通过该章的学习还应了解文档的作用与分类以及文档的管理与维护。

目 录

出版者的话
序 言
前 言
教学建议

第一部分 软件工程概述

第 1 章 软件与软件工程的概 念	2
1.1 软件的概念、特性和分类	2
1.1.1 软件的概念及特性	2
1.1.2 软件的分 类	4
1.2 软件危机与软件工程	5
1.2.1 软件危机	5
1.2.2 软件工程	5
1.3 系统工程的目标	6
1.4 软件生存期	7
1.5 软件生存期模型	8
1.5.1 瀑布模型	8
1.5.2 快速原型模型	10
1.5.3 增量模型	11
1.5.4 螺旋模型	12
1.5.5 喷泉模型	14
1.5.6 统一过程	14
1.6 软件工程知识体系及知识域	16
习 题	18
第 2 章 软件工程方法与工具	20
2.1 软件工程方法概述	20
2.1.1 传统方法	20
2.1.2 面向对象方法	21
2.1.3 形式化方法	21
2.2 面向对象的 概念与开发方法	22
2.2.1 对象	22
2.2.2 类与封装	23
2.2.3 继承	24

2.2.4 多态	25
2.2.5 消息通信	25
2.2.6 面向对象的软件开发方法	25
2.3 统一建模语言 UML 简介	26
2.3.1 UML 的产生和发展	27
2.3.2 UML 的特点	27
2.4 软件工具概述	28
2.4.1 软件工具的概念	28
2.4.2 软件工具的发展	28
2.4.3 软件工具的分类	28
2.4.4 常用软件工具介绍	29
2.5 UML 建模工具 Rational Rose	31
2.5.1 Rose 的特点	31
2.5.2 Rose 简介	32
2.5.3 Rose 的基本操作	33
习 题	36

第二部分 传统的软件开发方法

第 3 章 软件需求获取与结构化分析方法	38
3.1 需求获取与需求分析阶段的 任务	38
3.1.1 需求获取的任务和原则	38
3.1.2 需求获取的过程	39
3.1.3 软件需求分析阶段的 任务	42
3.2 结构化分析方法	43
3.2.1 功能建模	44
3.2.2 数据建模	48
3.2.3 行为建模	49
3.2.4 数据字典	52

3.2.5 加工规格说明	55	4.6.2 程序流程图	93
3.3 系统需求规格说明	57	4.6.3 N-S图	96
3.3.1 软件需求规格说明 模板	58	4.6.4 PAD图	97
3.3.2 SRS和DRD的质量 要求	59	4.6.5 伪代码	98
3.4 需求评审	61	4.6.6 自顶向下、逐步细化的 设计过程	99
3.4.1 正式的需求评审	61	4.7 软件设计规格说明	101
3.4.2 需求评审中的常见 风险	63	4.8 软件设计评审	104
3.5 需求管理	63	4.8.1 概要设计评审的检查 内容	104
3.5.1 需求跟踪	64	4.8.2 详细设计评审的检查 内容	105
3.5.2 需求变更管理	64	习题	106
习题	65	第5章 编码	109
第4章 结构化设计方法	66	5.1 程序设计语言	109
4.1 软件设计的概念及原则	66	5.1.1 程序设计语言的性能	109
4.1.1 软件设计的概念	66	5.1.2 程序设计语言的分类	111
4.1.2 软件设计的原则	66	5.1.3 程序设计语言的选择	113
4.2 结构化设计	69	5.2 程序设计风格	113
4.2.1 结构化软件设计的 任务	69	5.2.1 源程序文档化	114
4.2.2 结构化设计与结构化 分析的关系	69	5.2.2 数据说明标准化	115
4.2.3 模块结构及表示	70	5.2.3 语句结构简单化	115
4.2.4 数据结构及表示	74	5.2.4 输入/输出规范化	118
4.3 体系结构设计	75	5.3 编码规范	119
4.3.1 基于数据流方法的 设计过程	75	5.4 程序效率与性能分析	125
4.3.2 典型的数据流类型 和系统结构	75	5.4.1 算法对效率的影响	125
4.3.3 变换型映射方法	77	5.4.2 影响存储器效率的 因素	125
4.3.4 事务型映射方法	80	5.4.3 影响输入/输出的 因素	126
4.3.5 软件模块结构的 改进方法	82	习题	126
4.4 接口设计	89	第6章 软件测试方法	127
4.4.1 接口设计概述	89	6.1 软件测试的基本概念	127
4.4.2 人机交互界面	89	6.1.1 什么是软件测试	127
4.5 数据设计	91	6.1.2 软件测试的目的和 原则	128
4.5.1 文件设计	91	6.1.3 软件测试的对象	129
4.5.2 数据库设计	92	6.1.4 测试信息流	130
4.6 过程设计	92	6.1.5 测试与软件开发各阶段的 关系	131
4.6.1 结构化程序设计	92	6.1.6 白盒测试与黑盒测试	131
		6.2 白盒测试的测试用例设计	133

6.2.1	逻辑覆盖	133
6.2.2	语句覆盖	134
6.2.3	判定覆盖	134
6.2.4	条件覆盖	134
6.2.5	判定-条件覆盖	135
6.2.6	条件组合覆盖	136
6.2.7	路径覆盖	136
6.3	基本路径测试	137
6.4	黑盒测试的测试用例设计	141
6.4.1	等价类划分	141
6.4.2	边界值分析	144
6.5	软件测试的策略	146
6.5.1	单元测试	147
6.5.2	组装测试	149
6.5.3	确认测试	152
6.5.4	系统测试	154
6.5.5	测试的类型	154
6.6	人工测试	157
6.6.1	静态分析	157
6.6.2	人工测试方法	158
6.7	调试	159
	习题	160

第三部分 面向对象的软件开发方法

第7章	统一建模语言 UML 概述	162
7.1	UML 的基本模型	162
7.2	UML 的事物	162
7.2.1	结构事物	163
7.2.2	行为事物	163
7.2.3	分组事物	164
7.2.4	注释事物	164
7.3	UML 的关系	164
7.3.1	依赖关系	164
7.3.2	关联关系	165
7.3.3	泛化关系	168
7.3.4	实现关系	169
7.4	UML 的图	169
7.4.1	用例图	169
7.4.2	类图	171
7.4.3	交互图	173
7.4.4	状态图	174
7.4.5	活动图	175

7.4.6	实现图	176
	习题	178
第8章	面向对象分析	179
8.1	面向对象分析概述	179
8.1.1	面向对象分析的3个模型	179
8.1.2	对象模型的5个层次	179
8.2	建立用例模型	180
8.2.1	建立用例模型的过程	180
8.2.2	使用 Rose 创建用例模型	183
8.3	建立对象模型	187
8.3.1	划分主题	188
8.3.2	确定类与对象	188
8.3.3	确定关联	189
8.3.4	确定属性	189
8.3.5	确定服务	190
8.3.6	使用 Rose 建立类图	191
8.4	建立动态模型	197
8.4.1	顺序图	197
8.4.2	协作图	200
8.4.3	状态图	200
	习题	201
第9章	软件体系结构与设计模式	202
9.1	软件体系结构的基本概念	202
9.1.1	什么是体系结构	202
9.1.2	体系结构模式、风格和框架的概念	203
9.1.3	体系结构的重要作用	204
9.2	典型的体系结构风格	204
9.2.1	数据流风格	204
9.2.2	调用-返回风格	205
9.2.3	仓库风格	207
9.3	特定领域的软件体系结构	208
9.3.1	类属模型	208
9.3.2	参考模型	209
9.4	分布式系统结构	210
9.4.1	多处理器体系结构	210
9.4.2	客户机/服务器体系结构	211
9.4.3	分布式对象体系结构	214
9.4.4	代理	215

9.5	体系结构框架	216	10.8.3	重构对象设计模型	252
9.5.1	模型-视图-控制器	216	10.8.4	优化对象设计模型	252
9.5.2	J2EE 体系结构框架	217		习题	253
9.5.3	PCMEF 与 PCBMER 框架	218	第四部分 软件维护与软件管理		
9.6	设计模式	219	第 11 章 软件维护		
9.6.1	抽象工厂	220	11.1	软件维护的概念	256
9.6.2	单件	222	11.1.1	软件维护的定义	256
9.6.3	外观	222	11.1.2	影响维护工作量的 因素	257
9.6.4	适配器	223	11.1.3	软件维护的策略	257
9.6.5	职责链	225	11.2	软件维护活动	258
9.6.6	中介者	226	11.2.1	软件维护申请报告	258
9.6.7	观察者	228	11.2.2	软件维护工作流程	258
	习题	230	11.2.3	维护档案记录	259
第 10 章 面向对象设计			11.2.4	维护评价	259
10.1	面向对象设计过程与准则	231	11.3	程序修改的步骤及修改的 副作用	260
10.1.1	面向对象设计过程	231	11.3.1	分析和理解程序	260
10.1.2	面向对象设计准则	232	11.3.2	修改程序	260
10.2	体系结构模块及依赖性	233	11.3.3	修改程序的副作用 及其控制	262
10.2.1	类及其依赖性	233	11.3.4	重新验证程序	262
10.2.2	接口及其依赖性	236	11.4	软件的维护性	263
10.2.3	包及其依赖性	237	11.4.1	软件维护性定义	263
10.2.4	构件及其依赖性	238	11.4.2	软件维护性度量	264
10.3	系统分解	240	11.5	提高软件维护性的方法	265
10.3.1	子系统 and 类	240	11.5.1	使用提高软件维护性的 开发技术和工具	265
10.3.2	服务和子系统接口	240	11.5.2	实施开发阶段产品的 维护性审查	265
10.3.3	子系统分层和划分	240	11.5.3	改进文档	267
10.3.4	Coad & Yourdon 的 面向对象设计模型	241		习题	267
10.3.5	子系统之间的两种 交互方式	241	第 12 章 软件过程与软件过程改进		
10.3.6	组织系统的两种方案	242	12.1	软件过程概述	268
10.4	问题域部分的设计	242	12.2	软件生存期过程国际标准	270
10.5	人机交互部分的设计	244	12.3	软件过程成熟度	274
10.5.1	用户界面设计步骤	244	12.3.1	什么是软件过程 成熟度	274
10.5.2	Web 应用系统的 界面设计	245	12.3.2	过程制度化	275
10.6	任务管理部分的设计	246	12.4	软件能力成熟度模型	277
10.7	数据管理部分的设计	247	12.4.1	CMM 与 SEI	277
10.8	对象设计	248			
10.8.1	使用模式设计对象	248			
10.8.2	接口规格说明设计	251			

12.4.2	CMM 的演化	278	13.5.3	需求变更	325
12.4.3	CMM 族和 CMMI	279	13.5.4	需求变更控制	327
12.4.4	CMMI 1.2 简介	280	13.5.5	可追溯性管理	330
12.4.5	CMMI 评估	288	13.6	配置管理	331
12.5	软件过程改进	289	13.6.1	什么是软件配置 管理	332
12.5.1	软件过程改进的 IDEAL 模型	289	13.6.2	软件配置标识	332
12.5.2	软件过程改进框架	290	13.6.3	变更管理	334
12.5.3	有效的软件过程	292	13.6.4	版本控制	337
习题	293	13.6.5	系统建立	339
第 13 章	软件项目管理	294	13.6.6	配置审核	340
13.1	软件项目管理概述	294	13.6.7	配置状态报告	340
13.1.1	软件项目管理的目标	294	习题	341
13.1.2	软件项目管理涉及的 几个方面	294	第 14 章	软件工程标准及 软件文档	342
13.2	项目估算	296	14.1	软件工程标准	342
13.2.1	项目策划与项目 估算	296	14.1.1	标准的概念	342
13.2.2	软件规模估算的 功能点方法	297	14.1.2	软件标准化的意义	343
13.2.3	软件开发成本估算	301	14.1.3	标准的分类与分级	344
13.3	风险管理	307	14.1.4	软件工程标准的制定 与实施	347
13.3.1	什么是软件风险	307	14.1.5	软件组织内的标准化 工作	348
13.3.2	风险管理的任务	309	14.2	软件文档	348
13.3.3	风险评估	310	14.2.1	软件文档的作用 和分类	348
13.3.4	风险控制	313	14.2.2	软件基本文档的内容 要求	350
13.3.5	做好风险管理的建议	315	14.2.3	对文档编制的质量 要求	353
13.4	进度管理	316	14.2.4	文档的管理和维护	356
13.4.1	进度控制问题	316	习题	356
13.4.2	甘特图	318	参考文献		357
13.4.3	时标网状图	319			
13.4.4	PERT 图	319			
13.5	需求管理	321			
13.5.1	系统需求与软件需求	322			
13.5.2	需求工程	324			

第一部分
PART ONE

软件工程概述

软件与软件工程的概 念

计算机技术经过了 50 年的发展历程，取得了突飞猛进的发展。计算机的应用领域从单纯的科学计算发展到军事、经济、教育、文化等社会生产及生活的各个方面，推动了其他行业及领域的发展，改变了人们学习、工作及生活方式。进入 21 世纪，人类已从工业社会跨入了信息社会。

计算机软件系统是信息化的重要组成部分。计算机软件已形成了独立的产业，成为国民经济新的增长点和重要支柱。软件工程在软件开发中起着重要的作用，对软件产业的形成及发展起着决定性的推动作用。

20 世纪 90 年代，软件工程取得了突飞猛进的发展，软件工程已形成了一个比较完整的学科。

本章将对软件工程相关的概念、软件开发的过程及方法进行简要的介绍，使读者对软件工程的总体框架获得初步的了解。

1.1 软件的概念、特性和分类

1.1.1 软件的概念及特性

1. 软件的概念

我们国家 20 世纪 80 年代初的大学生知道软件的人并不多，甚至很多人从未听说过这个词，即使是当初软件专业毕业的学生也不曾想到软件的发展速度如此之快。今天的软件已无处不在，渗透到了各个行业之中。随着计算机大量进入家庭，计算机已经成为我们日常生活、学习和工作都离不开的工具，很多小学生已经开始接触和学习电脑。

虽然软件对于现代人来说并不陌生，但很多人对于软件的理解并不准确，“软件就是程序，软件开发就是编程序”这种错误观点仍然存在。因此，仍然有必要给软件一个明确的定义。

软件的一种公认的传统定义为：软件是计算机系统中与硬件相互依存的另一部分，软件包括程序、数据及其相关文档的完整集合。其中，程序是按事先设计的功能和性能要求执行的指令序列；数据是使程序能够正确地处理信息的数据结构；文档是与程序开发、维护和使用有关的图文资料。

在结构化程序设计时代，程序的最小单位是函数及子程序，程序与数据是分离的；在面向对象程序设计时代，程序的最小单位是类，在类中封装了相关的数据及指令代码。

2. 软件的特性

当今已有的人工制品数不胜数，然而计算机软件却与任何传统的制造业产品不同，它具有许多突出的特性，概括如下：