

<<软件工程>>

图书基本信息

书名：<<软件工程>>

13位ISBN编号：9787111339496

10位ISBN编号：7111339495

出版时间：2011-5

出版时间：瞿中、吴渝、常庆丽、等 机械工业出版社 (2011-05出版)

作者：瞿中，吴渝，常庆丽 等著

页数：357

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<软件工程>>

### 内容概要

《软件工程（第2版）》从实用的角度出发，根据教育部高教司主持编审的《中国计算机科学与技术学科教程2002》中对软件工程的要求组织编写，并参照美国ACM和IEEE Computing Curricula 2005教程关于软件工程的描述。

《软件工程（第2版）》详细介绍了软件工程、软件开发过程、软件计划、需求分析、总体设计、详细设计、编码、软件测试、软件维护、软件工程标准化和软件文档、软件工程质量、软件工程项目管理、开发实例等知识。

每章配有习题，以指导读者深入地进行学习。

《软件工程（第2版）》内容丰富，结构合理，既可作为高等院校计算机专业课程的教材或教学参考书，也可作为通信、电子信息、自动化等相关专业的计算机课程教材，还可供软件工程师、软件项目管理者 and 应用软件开发人员阅读参考。

## 书籍目录

出版说明前言第1章 概论1.1 软件的概念1.1.1 软件的定义以及特点1.1.2 软件技术的发展阶段1.1.3 软件的分  
类1.2 软件危机1.2.1 软件危机的定义1.2.2 软件危机产生的原因1.2.3 解决软件危机的途径1.3 软件工程  
1.3.1 软件工程的定义和研究对象1.3.2 软件工程的基本原理1.3.3 软件工程项目的基本目标1.3.4 软件工程  
的基本原则1.4 软件生存周期1.4.1 软件生存周期的概念1.4.2 软件开发工具1.5 软件开发过程模型1.6 软  
件开发方法简述1.7 软件工程的最新发展动向1.8 经典例题讲解小结习题第2章 结构化分析2.1 可行性研  
究2.1.1 问题定义2.1.2 可行性研究的任务2.1.3 可行性研究的步骤2.2 系统流程图2.2.1 系统流程图的符  
号2.2.2 系统流程图举例2.2.3 分层2.3 制订软件计划2.3.1 确定软件计划2.3.2 复审软件计划2.4 成本/效益分  
析2.4.1 成本估算技术2.4.2 成本/效益分析的方法2.5 需求分析的概念和任务2.5.1 需求分析的概念2.5.2 需  
求分析的层次2.5.3 需求分析的目标和任务2.5.4 需求分析的原则2.5.5 需求规格说明书2.5.6 评审2.6 获取  
需求的方法2.6.1 存在问题2.6.2 常用方法2.6.3 需求分析的过程2.6.4 结构化需求分析方法2.7 传统的软件  
建模2.7.1 软件建模2.7.2 数据模型的实体-联系图建立2.7.3 功能模型、行为模型的建立及数据字典2.7.4  
构建数据流图实例2.7.5 快速原型法分析实例2.8 经典例题讲解小结习题第3章 结构化设计3.1 总体设计  
的任务及过程3.1.1 总体设计的任务3.1.2 总体设计的过程3.2 总体设计的原理3.2.1 软件结构和过程3.2.2  
模块设计3.2.3 结构设计3.3 设计准则3.4 总体设计的常用方法及工具3.4.1 面向数据流的设计方法3.4.2 总  
体设计中的工具3.5 模块结构设计3.6 数据存储设计3.7 模型-视图-控制器框架3.7.1 MVC模式3.7.2 MVC  
中的模型类、视图类和控制器类3.7.3 MVC的实现3.8 总体设计说明书编写规范3.9 详细设计阶段的任  
务3.10 结构化详细设计的原则3.11 结构化详细设计的方法和工具3.11.1 详细设计的方法3.11.2 详细设计  
的工具3.11.3 详细设计工具的选择3.12 详细设计规格说明与复审3.12.1 详细设计说明3.12.2 设计复审3.13  
面向数据结构的结构化设计方法3.14 Jackson程序设计方法3.14.1 Jackson方法的基本思想3.14.2 Jackson结  
构图3.14.3 Jackson方法的设计技术3.15 Warnier程序设计方法3.15.1 Warnier方法的基本思想3.15.2 Warnier  
方法的设计技术3.16 基于组件的设计方法3.17 界面设计3.17.1 用户界面设计的一般原则和步骤3.17.2 字  
符界面设计3.17.3 菜单设计3.17.4 对话框设计3.17.5 多窗口界面设计3.18 软件体系结构3.18.1 软件体系结  
构的兴起3.18.2 软件体系结构的概念3.18.3 软件体系结构的现状及发展方向3.18.4 软件体系结构的描述  
方法3.19 软件体系结构与操作系统3.19.1 分层结构3.19.2 微内核结构3.20 经典例题讲解小结习题第4章 编  
码及测试4.1 程序设计语言4.1.1 程序设计语言的发展及分类4.1.2 选择程序设计语言的标准4.2 程序设计  
风格4.2.1 源程序文档化4.2.2 数据说明4.2.3 表达式和语句结构4.2.4 输入和输出4.3 程序效率4.4 编程安  
全4.5 结构化程序设计方法4.6 程序的复杂性及度量4.6.1 代码行度量法4.6.2 McCabe度量法4.6.3 Halstead  
度量法4.7 软件测试基础4.7.1 软件测试的意义4.7.2 基本概念4.7.3 软件测试的目的、任务、原则和研究  
对象4.7.4 软件测试的发展历史及趋势4.8 软件测试的方法4.8.1 静态测试和动态测试4.8.2 黑盒测试和白  
盒测试4.9 软件测试的步骤4.9.1 单元测试4.9.2 集成测试4.9.3 确认测试4.9.4 系统测试4.9.5 验收测试4.10 软  
件测试4.10.1 软件测试角色4.10.2 软件测试的需求规格说明4.10.3 软件测试设计说明4.11 测试设计和管理  
4.11.1 错误曲线4.11.2 测试用例设计4.12 软件测试工具4.12.1 自动软件测试的优点4.12.2 测试工具分  
类4.12.3 自动测试的相关问题4.13 经典例题讲解小结习题第5章 软件维护及软件再工程5.1 软件维护的概  
念5.1.1 软件维护的定义5.1.2 软件维护的分类5.1.3 软件维护成本5.2 软件维护的特点5.3 软件维护过程5.4  
软件维护的步骤5.5 软件的可维护性5.5.1 影响软件可维护性的因素5.5.2 软件可维护性度量5.5.3 提高软  
件的可维护性方法5.6 逆向工程和再工程5.6.1 预防性维护5.6.2 软件的逆向工程和再工程5.6.3 软件再工  
程的过程5.6.4 软件再工程的方法小结习题第6章 面向对象方法学6.1 面向对象方法学概述6.1.1 面向对象  
方法学的要点6.1.2 面向对象方法学的优点6.2 面向对象的概念6.2.1 对象6.2.2 其他概念6.3 面向对象建  
模6.4 对象模型6.4.1 类图的基本符号6.4.2 表示关系的符号6.5 功能模型6.5.1 用例图6.5.2 用例建模6.6 3种  
模型之间的关系6.7 经典例题讲解小结习题第7章 面向对象分析7.1 面向对象分析的基本过程7.1.1 概  
述7.1.2 三个子模型与五个层次7.2 需求陈述的书写7.3 建立对象模型7.3.1 确定类与对象7.3.2 确定关  
联7.3.3 划分主题7.3.4 确定属性7.3.5 识别继承关系7.3.6 反复修改7.4 建立动态模型7.5 建立功能模型7.5.1  
画出基本系统模型图7.5.2 画出功能级数据流图7.5.3 描述处理框功能7.6 定义服务7.7 经典例题讲解小结  
习题第8章 面向对象设计8.1 面向对象设计的准则8.1.1 模块化8.1.2 抽象化8.1.3 信息隐藏和封装8.1.4 对象  
的高内聚和弱耦合8.1.5 可扩充性8.1.6 可重用性8.2 启发规则8.2.1 设计结果应该清晰易懂8.2.2 一般——

## &lt;&lt;软件工程&gt;&gt;

特殊结构的深度应适当8.2.3 设计简单的类8.2.4 使用简单的协议8.2.5 使用简单的服务8.2.6 把设计变动减至最小8.3 软件重用8.3.1 概述8.3.2 类构件8.3.3 软件重用的效益8.4 系统分解8.4.1 子系统之间的两种交互方式8.4.2 组织系统的两种方案8.4.3 设计系统的拓扑结构8.5 设计问题域子系统8.6 设计人一机交互子系统8.6.1 设计人一机交互界面的准则8.6.2 设计人一机交互子系统的策略8.7 设计任务管理子系统8.7.1 确定事件驱动型任务8.7.2 确定时钟驱动型任务8.7.3 确定优先任务和关键任务8.7.4 确定协调任务8.7.5 确定资源需求8.8 设计数据管理子系统8.8.1 选择数据存储管理模式8.8.2 设计数据管理子系统8.9 设计类中的服务8.9.1 确定类中应有的服务8.9.2 涉及实现服务的方法8.10 设计关联8.10.1 关联的遍历8.10.2 实现单向关联8.10.3 实现双向关联8.10.4 链属性的实现8.11设计优化8.11.1 确定优先级8.11.2 提高效率的几项技术8.11.3 调整继承关系8.12经典例题讲解小结习题第9章 面向对象实现9.1 面向对象语言9.1.1 面向对象语言的优点9.1.2 面向对象语言的技术特点9.1.3 选择面向对象语言的原则9.2 面向对象程序设计风格9.2.1 提高可重用性9.2.2 提高可扩充性9.2.3 提高健壮性9.3 测试策略9.3.1 面向对象测试模型9.3.2 面向对象分析的测试9.3.3 面向对象设计的测试9.3.4 面向对象编程的测试9.3.5 面向对象的单元测试9.3.6 面向对象的集成测试9.3.7 面向对象的系统测试9.4 经典例题讲解小结习题第10章 软件工程标准化和软件文档10.1 软件工程标准化的概念10.1.1 什么是软件工程标准化10.1.2 软件工程标准化的意义10.1.3 软件工程标准化的类型10.2 软件工程标准的制定与推行10.3 软件工程标准的层次和体系框架10.3.1 软件工程标准的层次10.3.2 软件工程过程中版本控制与变更控制处理过程10.3.3 中国的软件工程标准化工作10.4 ISO9000国际标准简介10.5 软件文档10.5.1 软件文档的作用和分类10.5.2 对软件文档编制的要求10.5.3 软件文档的管理和维护10.6 经典例题讲解小结习题第11章 软件工程质量及项目管理11.1 软件质量特性11.1.1 软件质量的定义11.1.2 软件质量的特性11.2 软件质量的度量模型11.2.1 软件度量和软件质量的度量11.2.2 软件质量的度量模型11.3 软件质量保证11.3.1 什么是软件质量保证11.3.2 软件质量保证的主要任务11.3.3 软件质量保证策略11.4 技术评审11.5 软件质量管理体系11.5.1 软件产品质量管理的特点11.5.2 软件质量管理体系的指导思想11.5.3 软件质量管理体系11.6 软件项目管理11.6.1 软件项目管理的特点11.6.2 软件项目管理的主要职能11.6.3 软件项目管理的主要内容11.7 基于CASE技术的开发工具简介11.8 软件项目管理活动11.8.1 计划项目11.8.2 项目组织11.8.3 控制项目11.8.4 终结项目11.9 成本估算11.10计划和组织11.10.1 项目计划的制定11.10.2 项目组人员管理原则11.10.3 人员组织与管理11.11进度计划11.11.1 制定开发进度计划11.11.2 Gantt图与时间管理11.11.3 工程网络与关键路径11.11.4 项目进度跟踪与控制11.12风险管理11.12.1 风险识别与分类11.12.2 风险评估与分析11.12.3 风险策划与管理11.12.4 风险规避与监控11.13项目管理认证体系IPMP与PMP11.13.1 IPMP概况11.13.2 PMP简介11.13.3 我国目前的项目管理认证体系的发展状况11.14经典例题讲解小结习题第12章 开发实例12.1 项目论证和计划12.1.1 系统调查12.1.2 新系统的总体功能需求和性能要求12.1.3 系统开发的框架12.2 可行性分析12.3 需求分析12.3.1 数据流分析12.3.2 系统流程图12.3.3 数据字典12.4 总体设计12.4.1 功能模块图12.4.2 层次方框图12.4.3 IPO图12.4.4 系统的功能结构图12.4.5 人事管理工作的工作流程模型图12.4.6 系统数据库关系说明图12.5 详细设计12.6 系统实现12.7 测试与维护12.7.1 测试结果12.7.2 系统维护小结习题参考文献

## 章节摘录

版权页：插图：传统的软件是利用标准函数库技术来实现重用的，该方法试图用标准函数库中的函数作为“预制件”来建造新的软件系统。

但是，标准函数缺乏必要的“柔性”，不能适应不同应用场合的不同需要，并不是理想的可重用的软件成分。

在面向对象方法所使用的对象中，数据和操作是作为平等伙伴出现的。

因此，对象具有很强的自含性。

此外，对象固有的封装性和信息隐藏机制，使得对象的内部实现与外界隔离，具有较强的独立性。

由此可见，对象是比较理想的模块和可重用的软件成分。

面向对象的软件技术在利用可重用的软件成分构造新的软件系统时，有很大的灵活性。

以下两种方法可以重复使用一个对象类：一种方法是创建该类的实例，从而直接使用它；另一种方法是从它派生出一个满足当前需要的新类。

继承性机制使得子类不仅可以重用其父类的数据结构和程序代码，而且可以在父类代码的基础上方便地修改和扩充，这种修改并不影响对原有类的使用。

(4) 与人类习惯的思维方式一致面向对象的软件技术以对象为核心，用这种技术开发出的软件系统由对象组成。

对象是对现实世界实体的正确抽象，它是由描述内部状态表示静态属性的数据，以及可以对这些数据施加的操作，封装在一起所构成的统一体，对象之间通过传递消息互相联系，以模拟现实世界中不同事物彼此之间的联系。

面向对象的设计方法的基本原理是，使用现实世界的概念抽象地思考问题从而自然地解决问题。

它强调模拟现实世界中的概念而不强调算法，它鼓励开发者在软件开发的绝大部分过程中都用应用领域的概念去思考。

在面向对象的设计方法中，计算机的观点是不重要的，现实世界的模型才是最重要的。

面向对象的软件开发过程从始至终都围绕着建立问题领域的对象模型来进行。

对问题领域进行自然分解，确定需要使用的对象和类，建立适当的类等级，在对象之间传递消息实现必要的联系，从而按照人们习惯的思维方式建立起问题领域的模型，模拟客观世界。

在人的认识深化过程中，既包括了从一般到特殊的演绎思维过程，也包括了从特殊到一般的归纳思维过程。

人在认识和解决复杂问题时使用的最强有力的思维工具是抽象，也就是在处理复杂对象时，为了达到某个分析目的集中研究对象的与此目的有关的实质，忽略该对象的那些与此目的无关的部分。



版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>