

<<软件工程>>

图书基本信息

书名：<<软件工程>>

13位ISBN编号：9787111257400

10位ISBN编号：7111257405

出版时间：2009-1

出版时间：机械工业出版社

作者：李东生等著

页数：291

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>

## &lt;&lt;软件工程&gt;&gt;

## 前言

在高度信息化的21世纪,人们越来越认识到信息教育的重要性。人们都迫切希望信息教育能有较大发展。教育信息化也是摆在我们面前的重要任务。教育部明确要求高等教育实行信息化,要求在未来5年内实现信息化教育课程的数量达到15%~30%。信息社会离不开计算机技术,知识经济需要大量的计算机高级人才。我国正在加强计算机的高等教育,正着眼于为新世纪培养高素质的计算机人才,以适应信息社会高速发展的需要。

当前,全国各类高等院校都在各专业基础课程计划中增加计算机的课程内容,而作为与计算机科学密切相关的计算机、通信、信息等专业,更是在酝酿着教学的全面改革,以期规划出一整套面向21世纪的、具有中国高校计算机教育特色的课程计划和教材体系。

教育部《关于加强高等学校本科教育工作提高教育质量的若干意见》(教字【2001】4号)文件也强调指出:“要大力提倡编写、引进和使用先进教材。

教材的质量直接体现着高等教育和科学研究的发展水平,也直接影响本科教学的质量。

高等学校要结合学科、专业的调整,加快教材的更新换代。

”为推动高校教学改革,提高教学质量,我们重点抓了21世纪高等教育教学改革项目,组织并支持了“面向21世纪计算机系列教材规划”研究课题。

该课题组成员均由高校计算机系的专家教授组成。

他们有多年的丰富的教学经验,也具有很强的科研能力。

该课题的主要目标是密切结合国民经济的需要,优化计算机教材体系结构,力求将国际、国内计算机领域的新概念、新理论、新技术吸收到本系列教材中,编写出具有科学性、先进性、系统性、实用性、实践性很强的教材,经过推广使用,反复修改,不断提高。

“面向21世纪计算机系列教材规划”课题以编写非计算机专业的计算机课程、计算机专业的计算机网络课程、计算机软件课程三个系列教材为主要内容,计划在三年内出版13~16种书,服务于本科生、专科生、研究生,以及网络学院和软件学院的学生。

本课题把研究系列教材的重点放在影响和带动计算机学科发展的网络与软件,以及直接推动计算机普及和应用的非计算机专业三个方向上,目的是通过集中优势兵力,加强团队协作,能够在教材建设方面按系列有所突破。

相信本套教材的出版必将对教学改革和教材建设起到很大的推动和示范作用。

## &lt;&lt;软件工程&gt;&gt;

## 内容概要

《面向21世纪高等院校计算机系列教材·软件工程：原理、方法和工具》全面系统地介绍了软件工程的**概念、原理和技术**。

全书分3篇，第1篇（第1~8章）“面向过程的软件工程”，以传统的软件生存周期为主线，介绍软件工程的原理和方法，讲述了软件生存周期各阶段的任务、过程和方法。

第2篇（第9~16章）“面向对象的软件工程”，以UML语言为主线，结合当前流行的软件工程工具Rational Rose，介绍面向对象软件工程的原理和方法。

第3篇（第17章）“软件工程实验”，是在前两篇的理论知识基础上，介绍图书馆图书管理系统和浏览器系统实验。

《面向21世纪高等院校计算机系列教材·软件工程：原理、方法和工具》的特色是将软件工程理论与软件工程实际紧密结合，便于教学与应用，既可以作为高等院校计算机专业和软件工程专业本科生的教材，也可作为工程技术人员的参考书。

## 书籍目录

序前言第1篇 面向过程的软件工程第1章 软件危机、软件工程1.1 软件工程的发展史1.1.1 程序设计时代1.1.2 程序系统时代1.1.3 软件工程时代1.2 软件危机主要表现形式1.3 产生软件危机的原因及解决途径1.3.1 产生软件危机的原因1.3.2 解决软件危机的途径1.4 软件和软件工程1.4.1 软件1.4.2 软件工程1.5 软件质量1.6 软件的生存周期及开发模型1.6.1 软件生存周期1.6.2 软件开发模型1.7 习题第2章 可行性研究2.1 可行性研究的目的与任务2.2 可行性研究的步骤2.3 系统流程图2.3.1 系统流程图的符号2.3.2 系统流程图示例2.4 成本—效益分析2.4.1 货币的时间价值2.4.2 投资回收期2.4.3 纯收入2.5 可行性研究报告的主要内容2.6 习题第3章 软件需求分析3.1 需求分析的任务和步骤3.1.1 需求分析的任务3.1.2 需求分析的步骤3.2 需求获取的常用方法3.2.1 常规的需求获取方法3.2.2 快速建立软件原型来获取需求3.3 需求分析的方法3.3.1 功能分解方法3.3.2 结构化分析方法3.3.3 信息建模方法3.3.4 面向对象的分析3.4 结构化分析方法3.4.1 自顶向下逐层分解的分析策略3.4.2 结构化分析描述工具3.4.3 数据流图3.4.4 数据字典3.4.5 加工逻辑的描述3.5 需求分析图形工具3.5.1 层次方框图3.5.2 Wamier图3.5.3 IPO图3.6 SA方法的应用3.7 习题第4章 软件总体设计4.1 软件总体设计的目标和任务4.2 软件设计的概念和原理4.2.1 模块和模块化4.2.2 抽象4.2.3 信息隐蔽和局部化4.2.4 模块独立性及其度量4.3 软件结构设计准则4.4 软件结构设计的图形工具4.4.1 软件结构图4.4.2 层次图4.4.3 HIPO图4.5 结构化设计方法4.5.1 数据流图的类型4.5.2 结构化设计方法的步骤4.5.3 变换型分析设计4.5.4 事务型分析设计4.6 习题第5章 软件详细设计5.1 详细设计的目的与任务5.2 结构化程序设计5.3 详细设计工具5.3.1 程序流程图5.3.2 N-S图5.3.3 PAD图5.3.4 过程设计语言5.4 习题第6章 软件编码6.1 程序设计语言的分类6.1.1 基础语言6.1.2 结构化语言6.1.3 面向对象的语言6.2 程序设计语言的选择6.3 程序设计风格6.3.1 程序内部文档6.3.2 数据说明6.3.3 语句构造6.3.4 输入/输出6.3.5 效率6.4 习题第7章 软件测试7.1 软件测试的目标7.2 软件测试的原则7.3 软件测试方法7.3.1 静态测试与动态测试7.3.2 黑盒测试法与白盒测试法7.4 软件测试用例的设计7.4.1 白盒技术7.4.2 黑盒技术7.5 软件测试过程7.5.1 单元测试7.5.2 集成测试7.5.3 确认测试7.5.4 系统测试7.6 调试7.6.1 调试的目的7.6.2 调试技术7.7 习题第8章 软件维护8.1 软件维护的分类8.2 软件维护的特点8.2.1 结构化维护与非结构化维护8.2.2 维护的代价8.2.3 软件维护中存在的问题8.3 软件可维护性8.3.1 软件可维护性的定义8.3.2 软件可维护性的度量8.3.3 提高软件可维护性的方法8.4 维护的副作用8.5 软件再工程8.5.1 软件再工程与逆向工程的概念8.5.2 实施软件再工程的原因8.5.3 软件再工程技术8.6 习题第2篇 UML与面向对象的软件工程第9章 UML简介9.1 UML概述9.1.1 UML的组成9.1.2 UML的特点和用途9.1.3 UML的模型视图简介9.2 UML软件开发工具简介9.2.1 RationalRose9.2.2 Visio简介9.3 习题第10章 面向对象的概念10.1 面向对象的方法10.1.1 面向对象建模10.1.2 面向对象的方法与传统软件方法的比较10.2 对象与类及其UML表示10.2.1 对象10.2.2 类与实例10.2.3 对象属性与操作10.2.4 对象类的关联10.3 聚集、组合、继承和多态10.3.1 聚集与组合10.3.2 抽象与继承10.3.3 多态10.4 习题第11章 对象设计模式11.1 对象设计模式概念11.1.1 历史背景11.1.2 对象设计模式11.1.3 设计模式的分类11.2 几种典型的对象设计模式及应用11.2.1 行为型模式中的职责键模式及应用11.2.2 结构型模式中的外观模式及应用11.2.3 创建型模式中的抽象工厂模式及应用11.3 对象类的高级概念11.3.1 抽象类、参数对象类、型与实现对象类11.3.2 接口11.3.3 版型11.4 组件、包和结点11.4.1 组件11.4.2 包和包图11.4.3 结点11.5 习题第12章 面向对象的软件开发过程12.1 RUP概述12.1.1 RUP的历史12.1.2 RUP的特色12.2 RUP软件开发生命周期和建模12.2.1 RUP软件开发生命周期12.2.2 RUP的动态结构12.2.3 RUP的静态结构12.2.4 RUP的建模12.3 面向对象软件开发过程的案例分析12.3.1 系统需求12.3.2 系统的静态结构模型12.3.3 顺序图和协作图12.3.4 状态图和活动图12.3.5 组件图和部署图12.4 习题第13章 软件开发工具Rose13.1 Rose的用例图与类图13.1.1 用例图13.1.2 类图13.2 Rose的交互图与状态机图13.2.1 交互图13.2.2 状态机图13.3 包、组件图与部署图13.3.1 包13.3.2 组件图13.3.3 部署图13.4 Rose的代码生成和逆向工程13.4.1 代码生成13.4.2 逆向工程13.5 Rose应用举例13.6 习题第14章 Rose及其应用14.1 Rose的主要功能14.2 Rose的使用14.2.1 Rose主菜单窗口14.2.2 模型与工作方式的组织14.3 教学管理系统的分析与设计14.3.1 系统需求分析14.3.2 系统问题领域分析14.3.3 静态结构模型的建立14.3.4 动态行为模型的建立14.3.5 物理模型的建立14.4 UML类图与VB代码的转换14.4.1 VB代码生成属性14.4.2 Rose模型-VB代码的生成14.4.3 VB代码的逆向工程14.5 UML类图与Java代码的转换14.5.1 UML类图-Java代码的生成14.5.2 Java代码的逆向工程14.6 习题第15章 面向对象的软件体系结构15.1 软件体系

## &lt;&lt;软件工程&gt;&gt;

结构概述15.2 流程处理与客户机/服务器体系结构15.2.1 流程处理15.2.2 客户机 / 服务器体系结构15.3 三层和多层体系结构15.3.1 三层体系结构15.3.2 浏览器 / 服务器体系结构风格15.3.3 多层体系结构15.3.4 团聚与串行15.4 公共对象请求代理体系结构15.5 基于层次消息总线的体系结构风格15.6 异构结构风格15.7 习题第16章 软件工程新技术16.1 软件复用技术16.1.1 软件复用概念及分类16.1.2 软件复用的关键技术和复用粒度16.2 中间件技术16.2.1 中间件概念及特点16.2.2 中间件的分类16.3 组件技术16.3.1 组件与组件化16.3.2 组件模型及描述语言16.3.3 组件的检索与组装16.4 计算机辅助软件工程技术16.4.1 CASE的基本概念16.4.2 CASE工具与集成CASE环境16.5 软件产品线技术16.5.1 软件产品线基本概念16.5.2 软件产品线方法16.5.3 北大青鸟工程16.6 软件过程与标准化16.6.1 软件过程及其改进16.6.2 ISO9000标准16.6.3 软件能力成熟度模型16.7 习题第3篇 软件工程实验第17章 软件工程实验大纲17.1 面向过程的软件工程实验17.1.1 实验1 图书馆图书管理系统的可行性分析17.1.2 实验2 图书馆图书管理系统的项目开发计划17.1.3 实验3 图书馆图书管理系统的需求分析17.1.4 实验4 图书馆图书管理系统的总体设计17.1.5 实验5 图书馆图书管理系统的详细设计及编码实现17.1.6 实验6 图书馆图书管理系统的系统测试17.1.7 实验7 图书馆图书管理系统用户手册的撰写17.1.8 实验8 图书馆图书管理系统项目开发总结报告的撰写17.2 面向对象的软件工程实验17.2.1 实验1 浏览器系统的可行性分析17.2.2 实验2 浏览器系统的项目开发计划17.2.3 实验3 浏览器系统的需求分析17.2.4 实验4 浏览器系统的体系结构设计17.2.5 实验5 采用面向对象方法进行浏览器系统的详细设计17.2.6 实验6 利用RationalRose正向工程功能完成编码的实验17.2.7 实验7 利用RationalRose逆向工程完成修改设计的实验17.2.8 实验8 浏览器系统的测试17.2.9 实验9 浏览器系统用户手册的撰写17.2.10 实验10 浏览器系统项目开发总结报告的撰写附录附录A UML图总结附录B UML中定义的常用版型、约束和标记附录C GOF给出的软件设计模式参考文献

## 章节摘录

第1篇 面向过程的软件工程 第1章 软件危机、软件工程 “软件工程”一词是由北大西洋公约组织（North Atlantic Treaty Organization, NIATO）的计算机科学家在联邦德国召开的国际会议上首次提出来的。

产生软件工程这门学科的时代背景是“软件危机”。

软件工程的发展和应用不仅缓和了软件危机，而且促使一门新兴的工程学科诞生了。

1.1 软件工程的发展史 软件工程是随着计算机系统的发展而逐步形成的计算机科学领域中一门新兴的学科。

软件工程的发展经过了三个时代。

1.1.1 程序设计时代 1946年到1956年为程序设计时代，在这个时代，人们用很大力气研究和发​​展计算机硬件，经历了从电子管计算机到晶体管计算机的变革，然而对计算机软件的研究和发展却不够重视。

当时，由于硬件的价格昂贵，运行速度低，内存容量小，所以，当时的程序员非常强调“程序设计技巧”，把缩短每一个微秒的CPU时间和节省每一个二进制存储单元，作为程序设计的重要目标。

但设计的程序难读、难懂、难修改。

1.1.2 程序系统时代 1956年到1968年为程序系统时代，或者称为“程序+说明”时代。

在这个时代，硬件经历了从晶体管计算机到集成电路计算机的变革，CPU速度和内存容量都有了很大的提高，从而为计算机在众多领域中的应用提供了潜在的可能性。

这个时代的另一个重要特征是出现了“软件作坊”。

这是因为随着计算机应用的普及和深化，需要的软件往往规模相当庞大，以致单个用户无法开发，此外，许多不同的部门和企业往往需要相同或者类似的软件，各自开发就会浪费很大的人力。

在这种形势下，“软件作坊”就应运而生了。

不过这个时代的开发方法基本上沿用了程序设计时代的开发方法，但开始提出了结构化的方法。

随着计算机应用的日益普及，软件需求量急剧增长，用户的需求和使用环境发生变化时，软件可修改性很差，往往需要重新编制程序，其研制时间很长，不能及时满足用户要求，质量得不到保证，开发人员的素质和落后的开发技术不适应规模大、结构复杂的软件开发，因此，产生了尖锐的矛盾，所谓的“软件危机”便由此产生了。

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>