

<<软件工程>>

图书基本信息

书名：<<软件工程>>

13位ISBN编号：9787508475288

10位ISBN编号：7508475283

出版时间：2010-6

出版时间：水利水电出版社

作者：景秀丽，吕洪林 主编

页数：268

字数：426000

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

软件工程是一门运用工程学的原理和方法来组织和管理软件的开发、运行和维护，力求以高效率生产出高质量的软件产品的学科。

软件工程是计算机科学技术领域的一个重要分支，在软件开发实战中起到基础指导的作用。

当今，国内软件工程教材较多，相关内容的深浅度、侧重点各有不同。

我们依据高等院校本科生“软件工程”学科教学大纲所规定的教学要求编写了本教材，同时把多年来软件工程的教學经验和教學实践成果融入到书中，介绍了软件工程的基础理论、软件开发流程以及相关实践操作，从而指导学生进行软件开发活动。

本教材也适用于高职高专计算机相关专业和软件技术相关专业的学生使用。

在教材编写上，本着深入浅出的原则，注重内容的先进性、系统性和实用性，力求反映软件开发技术的最新成果。

在结构安排上，通俗易懂地阐述软件工程的基础理论知识，循序渐进，注重结合实践内容。

在每章内容后面均附有小结和课后习题。

本书共14章，针对计算机专业和软件技术专业学生的实际特点设计了知识结构，全面系统地阐述了软件工程的知识体系、软件开发的相关流程，具体分布如下：第1章，软件工程概述。

主要介绍了软件工程的概念、软件生命周期和软件开发模型。

第2章，软件可行性研究。

概括描述了可行性研究的任务、可行性研究的步骤和可行性研究的报告。

## <<软件工程>>

### 内容概要

本书详细地阐述了软件工程基础知识及其相关的实用技术。

内容包括软件工程概述、软件可行性研究、软件需求分析、软件总体设计、软件详细设计、面向对象技术、统一建模语言UML、程序编码、软件测试、软件实施与维护、软件重用技术、软件项目计划与管理、软件开发工具与环境以及软件开发实战(物流网络管理系统设计与实现)。

本书注重软件工程基本知识和基本概念的形象表述,注重内容系统性与实用性的恰当结合,力求做到通俗易懂,突出实践性。

教材通过对软件工程常用方法的介绍,展现软件设计的实际运作过程,帮助读者掌握相关知识并在软件工程项目的开发上使用工程化标准。

本书适合高等院校本科层次和高职层次的计算机和软件技术相关专业的学生使用,也可作为软件开发技术培训的教材,同时还可作为从事软件开发人员的参阅资料。

## &lt;&lt;软件工程&gt;&gt;

## 书籍目录

前言	第1章 软件工程概述	1.1 软件概述	1.1.1 软件的概念	1.1.2 软件分类
	1.1.3 软件的发展	1.1.4 软件危机	1.2 软件工程的定义和原理	1.2.1 软件工程的定义和原理
	1.2.2 软件的目标	1.2.3 软件的原则	1.3 软件生命周期	1.4 软件开发模型
	1.4.1 瀑布模型	1.4.2 快速原型模型	1.4.3 增量模型	1.4.4 螺旋模型
	1.4.5 喷泉模型	1.5 小结	1.6 习题	第2章 软件可行性研究
	2.1 可行性研究的任务	2.2 可行性研究的步骤	2.3 可行性研究报告	2.4 小结
	2.5 习题	第3章 软件需求分析	3.1 需求分析概述	3.1.1 需求分析的任务
	3.1.2 需求分析的步骤	3.1.3 软件需求分析的原则	3.2 面向数据流的分析方法	3.2.1 基于数据流的分析方法
	3.2.2 数据流图	3.2.3 数据字典	3.2.4 加工逻辑说明	3.3 需求分析方法与图形工具
	3.4 实体—联系图	3.4.1 数据对象、属性与关系	3.4.2 实体 - 联系图和实体模型	3.5 需求规格说明与评审
	3.6 小结	3.7 习题	第4章 软件总体设计	4.1 总体设计概述
	4.1.1 总体设计的任务	4.1.2 总体设计的步骤	4.2 总体设计的图形描述工具	4.2.1 层次图
	4.2.2 HIPO图	4.2.3 结构图	4.3 模块化设计	4.3.1 模块化与局部化
	4.3.2 模块独立性	4.3.3 抽象与信息隐蔽	4.4 面向数据流的设计方法	4.4.1 基本概念
	4.4.3 事务分析	4.4.4 设计优化原则	4.5 小结	4.6 习题
	第5章 软件详细设计	5.1 详细设计概述	5.1.1 详细设计的任务	5.1.2 详细设计的步骤
	5.2 详细设计的图形描述工具	5.2.1 程序流程图	5.2.2 N-S图	5.2.3 PAD图
	5.2.4 过程设计语言PDL	5.2.5 判定表和判定树	5.3 Jackson设计方法	5.3.1 Jackson方法概述及其图例
	5.3.2 Jackson程序设计过程	5.4 Warnier设计方法	5.4.1 Warnier方法概述及其图例	5.4.2 Warnier程序设计过程
	5.5 小结	5.6 习题	第6章 面向对象技术	6.1 面向对象技术的概述
	6.1.1 面向对象的基本概念	6.1.2 面向对象技术的优势	6.2 面向对象的开发模型	6.3 面向对象的分析
	6.3.1 论域分析	6.3.2 应用分析	6.4 面向对象的系统设计	6.4.1 系统设计过程
	6.4.2 子系统设计	6.4.3 人机交互设计	6.4.4 任务管理设计	6.4.5 数据管理设计
	6.5 面向对象的实现	6.5.1 程序设计语言	6.5.2 类和应用程序的实现	6.6 小结
	6.7 习题	第7章 统一建模语言(UML)	7.1 UML概述	7.1.1 UML概念
	7.1.2 UML的演变	7.1.3 UML的主要内容	7.1.4 UML的应用	7.2 UML的概念模型
	7.2.1 UML的构造块	7.2.2 UML的规则	7.2.3 UML的公共机制	7.3 UML的静态建模机制
	7.3.1 用例模型	7.3.2 类和对象模型	7.3.3 包	7.4 UML的动态建模机制
	7.4.1 消息	7.4.2 状态图	7.4.3 时序图	7.4.4 协作图
	7.4.5 活动图	7.5 UML的物理架构建模	7.5.1 逻辑架构与物理架构	7.5.2 构件图和配置图
	7.6 小结	7.7 习题	第8章 程序编码	8.1 程序设计语言
	8.1.1 程序设计语言的分类	8.1.2 程序设计语言的特点	8.1.3 程序设计语言的选择	8.2 结构化程序设计
	8.3 程序设计风格	8.3.1 源程序文档化	8.3.2 数据说明方式	8.3.3 语句构造方法
	8.3.4 输入/输出技术	8.4 程序设计效率	8.5 程序复杂性度量	8.5.1 代码行度量法
	8.5.2 McCabe度量法	8.5.3 HalStead方法	8.6 小结	8.7 习题
	第9章 软件测试	9.1 软件测试的基本概念	9.1.1 软件测试的定义	9.1.2 软件测试的原则
	9.2 软件测试方法	9.2.1 静态测试与动态测试	9.2.2 黑盒测试	9.2.3 白盒测试
	9.3 软件测试流程	9.3.1 软件测试总体流程	9.3.2 软件测试的具体策略	9.4 测试用例的设计
	9.5 面向对象软件测试	9.6 软件测试相关文档	9.7 小结	9.8 习题
	第10章 软件实施与维护	10.1 软件产品的实施	10.1.1 软件产品实施概述	10.1.2 软件产品实施过程
	10.2 软件产品的维护活动	10.3 软件维护过程	10.4 软件维护文档	10.5 软件可维护性
	10.5.1 决定软件可维护性的因素	10.5.2 软件可维护性的度量	10.5.3 提高软件可维护性的方法	10.6 软件维护的深化——软件再工程
	10.7 小结	10.8 习题	第11章 软件重用技术	11.1 软件重用技术概述
	11.1.1 软件重用定义	11.1.2 软件重用形式	11.1.3 软件重用分类	11.1.4 软件复

## &lt;&lt;软件工程&gt;&gt;

用的现状和流行的软件重用技术 11.2 基于构件的软件开发 11.2.1 可重用软件构件的开发  
11.2.2 可重用软件构件的组织 11.2.3 可重用软件构件的分类和检索 11.3 面向对象的  
软件重用技术 11.4 小结 11.5 习题 第12章 软件项目计划与管理 12.1 软件项目的  
计划与组织 12.1.1 软件开发的进度计划 12.1.2 软件开发的组织机构 12.1.3 软件  
人员配备 12.2 软件成本估算及控制 12.3 软件工程标准与软件文档 12.3.1 软件工程标  
准 12.3.2 软件文档 12.4 小结 12.5 习题 第13章 软件开发工具与软件工程环境  
13.1 软件开发工具 13.1.1 软件开发工具的功能 13.1.2 常用软件开发工具介绍 13.2  
软件工程环境 13.2.1 软件工程环境的概念 13.2.2 软件开发环境的特点 13.3  
CASE技术 13.3.1 CASE定义 13.3.2 CASE环境的组成与结构 13.4 小结 13.5 习  
题 第14章 软件开发实战——物流网络管理系统设计与实现 14.1 概述 14.2 系统分析  
14.3 总体设计 14.4 详细设计 14.5 系统实现 14.6 系统测试 参考文献

## &lt;&lt;软件工程&gt;&gt;

## 章节摘录

插图：3.工具软件工具软件是20世纪80年代发展起来的，它是系统软件和应用软件之间的支持软件。一般用来辅助和支持开发人员开发和维护应用软件，以提高软件的开发质量和生产率。

它包括需求分析工具、设计工具、编码工具、测试工具、维护工具和管理工具等。

工具软件又可分为垂直工具软件和水平工具软件。

垂直工具软件是指生命周期的某一阶段特定活动所使用的工具软件，如分析、设计、测试等活动；水平工具软件是指整个生命周期活动所使用的工具软件，如项目管理、配置管理等活动。

4.可重用软件可重用技术是最近几年提出来的。

实际上过去就有这种技术，如各种标准程序库，通常它是计算机厂家提供的系统软件中的一部分，对这些标准程序库里的标准子程序稍加改造，甚至不经改造就可以把它们编入新开发的程序。

但过去的这种标准程序应用面比较窄，大多只限于一些数学子程序。

今天，世界已把可重用范围扩展到算法以外，数据结构也可以重用。

20世纪90年代的可重用构件则是把数据和相应的操作两者封装在一起（通常叫作类或对象），使软件工程师能够用可重用构件来建立新的应用程序。

例如，现在的交互式界面一般就是用这种可重用构件组成的。

这些可重用构件能够建立图形窗口、下拉菜单以及各种交互机制。

建立这样的界面所需要的数据结构和处理细节都包含在一个由界面构件所组成的可重用库里。

<<软件工程>>

编辑推荐

《软件工程》：21世纪高等院校创新精品规划教材

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>