<<软件工程实用教程>>

图书基本信息

书名:<<软件工程实用教程>>

13位ISBN编号:9787302275688

10位ISBN编号: 7302275688

出版时间:2012-9

出版时间:清华大学出版社

作者: 陶华亭编

页数:288

字数:480000

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

<<软件工程实用教程>>

内容概要

《软件工程实用教程(第2版)》在第1版的基础上不仅补充了"软件工程"学科的较新内容,又 在体系结构上进行了调整,使内容和形式都更适用于教学。

"项目管理"和"工程技术"是软件工程两大部分内容,为强调项目管理的重要性,《软件工程实用 教程(第2版)》打破了传统教材的内容体系模式,把"项目管理"内容安排在前面,这样就可以在 一个学期的前半部分只进行课堂教学。

第1章介绍基本理论与概念;第2章介绍软件开发过程,第3章介绍软件项目管理;第4~9章基于结构化方法,介绍了软件工程技术、方法、工具和软件工程思想;第10~16章介绍面向对象的方法,重点结合案例介绍了基于uml的分析与设计过程,强调培训面向对象的分析和设计技能;附录部分补充了rational

rose简介和课程教学计划及试题样卷。

《软件工程实用教程(第2版)》案例完整,内容详实,可操作性强,注重对学生实践能力的培养,适合作为高等院校计算机和相关专业"软件工程"课程的教材。

<<软件工程实用教程>>

书籍目录

第1章概述

- 1.1软件工程学科的诞生
- 1.2基本概念和原理
- 1.3软件工程的主要内容
- 1.4软件工程师职业道德规范与要求
- 本章小结
- 习题

第2章软件过程

- 2.1软件过程的概念
- 2.2常见的软件过程模型
- 2.3软件过程的新发展
- 2.4敏捷方法
- 2.5rup
- 2.6软件过程改进与cmm
- 本章小结
- 习题
- 第3章软件项目管理
- 3.1概述
- 3.2软件规模估算
- 3.3工作量估算
- 3.4开发周期估算
- 3.5估算方法
- 3.6项目进度计划与控制
- 3.7人员组织
- 3.8质量保证
- 3.9软件配置管理
- 本章小结
- 习题
- 第4章项目调查与初步计划
- 4.1项目调查的任务
- 4.2项目调查的步骤
- 4.3结构化方案设计
- 4.4面向对象的方案设计
- 4.5成本效益分析
- 4.6可行性分析报告
- 本章小结
- 习题
- 第5章结构化需求分析
- 5.1需求分析概述
- 5.2结构化分析方法
- 5.3数据流图的绘制
- 5.4编写数据字典
- 5.5加工逻辑的分析与表达
- 5.6需求验证与评审
- 本章小结

<<软件工程实用教程>>

$\overline{}$	日志
~	疋火

第6章结构化设计

- 6.1总体设计的过程
- 6.2设计原理
- 6.3启发性设计原则
- 6.4描绘软件结构的图形工具
- 6.5面向数据流的设计方法
- 6.6实例

本章小结

习题

第7章详细设计

- 7.1代码设计
- 7.2数据库设计
- 7.3规范化理论
- 7.4数据库设计实例
- 7.5系统平台设计
- 7.6对话设计
- 7.7输入输出设计
- 7.8处理过程设计
- 7.9编写系统设计说明书
- 本章小结
- 习题

第8章编程与测试

- 8.1概述
- 8.2编程
- 8.3软件测试基本概念
- 8.4白盒法
- 8.5黑盒法
- 8.6调试
- 8.7单元测试
- 8.8集成测试
- 8.9确认测试
- 8.10软件可靠性

本章小结

习题

第9章软件维护

- 9.1软件维护的内容
- 9.2软件维护的特点
- 9.3软件维护的实施
- 9.4软件的可维护性

本章小结

习题

第10章面向对象方法

- 10.1软件工程方法论
- 10.2从结构化到面向对象
- 10.3面向对象的基本概念

10.4uml

<<软件工程实用教程>>

- 10.5uml提供的常用图
- 10.6面向对象的建模过程
- 本章小结
- 习题
- 第11章用例模型
- 11.1用例模型简介
- 11.2提取用例
- 11.3画用例图
- 11.4编写用例报告
- 11.5构造用户界面原型
- 本章小结
- 习题
- 第直2章全局分析
- 12.1选择构架模式
- 12.2识别关键抽象
- 12.3标识"关键技术"
- 12.4选定分析局部
- 本章小结
- 思考题
- 第13章局部分析
- 13.1提取"分析类"
- 13.2转述需求场景
- 13.3整理分析类
- 13.4案例
- 本章小结
- 习题
- 第14章全局设计
- 14.1确定核心元素
- 14.2引入外围元素
- 14.3优化模型结构
- 本章小结
- 习题
- 第15章局部设计
- 15.1实现需求场景
- 15.2实现子系统接口
- 本章小结
- 习题
- 第16章细节设计
- 16.1精化 "属性"与"操作"
- 16.2明确类之间的关系
- 本章小结
- 习题
- 附录a建模工具rational rose简介
- 附录b软件工程职业道德规范和实践要求(5.2版)
- 附录c参考教学计划
- 附录d试题样卷
- 参考文献

<<软件工程实用教程>>

<<软件工程实用教程>>

章节摘录

版权页: 插图: 不难发现,程序结构简单多了。

下面分析一下这两种情况之间的区别。

第一种情况,是把本该每个人都各负其责的事情都大包大揽到自己身上,把所有人该考虑的事情都放在一起考虑,很艰难地按照做事的流程整理出头绪,然后,试图井井有条地去完成它。

而与此同时,除你之外的其他人都无事可做,等着你把事情做得滴水不漏。

这种做法是,先把问题糅合在一起,再分解问题、理出头绪,一个人包揽一切,显然你得是一个"超人"。

但是,在结构化编程的思维模式下,大家又都自然而然地去这样创造了一个又一个的"超人"。

不言而喻,要把交织在一起的事情,事无巨细、通盘考虑得井井有条,程序的结构就会设计得非常复杂。

这样的"超人"不容易设计,也不容易维护。

在第二种情况下,事情就变得简单多了。

你只给出通用的提示,然后期待每个人能自己承担起自己的责任,弄清怎样完成本该自己完成的任务

两者比较,最大的区别是,你认识到了"软件功能是由若干个对象,相互协作来实现的,每个对象都 是一个有行为能力的、责任明确的软件要素"。

这里有一个"责任的转移",在第一种情况下,你要对一切负责,在第二种情况下,学生对自己负责

两种情况下,要实现的目标相同,但"软件结构"差异很大,其中的奥妙就在于:责任从你转移到了 每个人。

不难发现,为了控制问题的复杂性,两种方法论都使用了"分解"的手法,结构方法是"分解功能,包揽一切",面向对象的方法是"分解责任,各负其责"。

前者是把问题糅合在一起,越搞越复杂,自己给自己找麻烦;后者则是明确责任,知人善任,化繁为简。

从结构化到面向对象,软件设计的任务就变成了: (1)寻找这样的"责任人"(类或对象); (2)完善其应该具备的特征(属性、数据); (3)赋予它行为能力(方法或操作),让其承担起自己的那一份责任。

这种方法的思维模式是: (1) 关注要解决的问题可以划分为哪些责任; (2) 找到能够负担指定责任的"人"(寻找类或对象); (3) 对初步找到的"责任人"进行排查,让它们参与实现相关的功能,通过类和对象的交互场景,归纳每个对象应该负有的全部责任; (4) 然后,赋予"责任人"特征或行为能力(设计类的数据和操作),通过消息传递,让其各自承担起责任。

第二种方案除了责任明确、解决问题简单以外,还具有能够应对需求变更的先天优势。

为了看到这种责任重新安排带来的影响,我们考虑一下在有了新的需求变化时,情况会怎样。

假设,要求你通知听讲的学生中的研究生,他们可能需要在听完你的讲座之后,给本讲座作一个书面评价,并交到教务处。

在第一种情况下,我们将不得不对控制程序进行结构修改,以区别研究生和本科生,然后给研究生特殊指示。

而在第二种情况下,每个人各司其职,只需要从"学生"中再派生出一个"研究生"类,赋予它相应的责任就行了。

而作为一个特殊类型对象的你,除了需要说"找到你们下一个听讲的教室"外,只需再多说一句话: "请研究生对本讲座进行书面评价,并交到教务处"。

每个人仍然只需各行其是,程序的结构无须改动,只是为你这个特殊的类添加了责任和新创建了一个 研究生类。

<<软件工程实用教程>>

编辑推荐

《21世纪普通高校计算机公共课程规划教材:软件工程实用教程(第2版)》案例完整,内容详实,可操作性强,注重对学生实践能力的培养,适合作为高等院校计算机和相关专业"软件工程"课程的教材。

<<软件工程实用教程>>

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