

<<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：9787111318446

10位ISBN编号：7111318447

出版时间：2011-1

出版时间：机械工业出版社

作者：吕云翔，王洋，王昕鹏 编著

页数：231

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>>

前言

20世纪60年代,为了解决出现的“软件危机”,人们提出了软件工程的观念,并将其定义为“为了经济地获得可靠的和能在实际机器上高效运行的软件,而建立和使用的健全的工程规则”。

经过40多年的发展,人们对软件工程有了更全面更科学的认识,软件工程已经成为一门包括理论、方法、过程等内容的独立的学科,并出现了相应的软件工程支撑工具。

然而即使在21世纪的今天,软件危机的种种表现依然没有彻底地得到解决,现实中的很多项目依然挣扎在无法完成或无法按照规定的时间、成本、按预期的质量完成的泥潭中,面临着失败的危险。

究其原因,是软件工程的思想和方法并未深入到计算机科学技术,特别是软件开发领域中,并指导人们的开发行为。

为了振兴中国的计算机和软件产业,培养具备软件工程思想和技术,并具有相应开发经验的人才,国家近年来十分重视软件工程相关课程的建设 and 人才培养。

除了建立专门的软件工程专业,也倡导在计算机科学与技术及相关专业开设软件工程课程,使得软件工程思想和技术在中国的IT人才中得到普及。

本书面向普通大学计算机及相关专业的学生,也可供计算机和软件开发爱好者自学使用。

为了体现软件工程知识体系的层次,并具有更好的实践指导意义,本书除了介绍软件工程的基本概念、理论、方法和过程,还十分重视软件工程相关工具使用方法的介绍,并通过实际案例来讲述软件工程在实际软件项目开发中的应用和体现。

本书按照典型的软件开发过程来组织内容,全书分为8章。

第1章概要介绍软件工程的起源,软件工程相关概念,软件工程方法、过程和工具;第2~5章分别介绍软件可行性研究及软件需求分析,软件设计,软件编码及实现,软件测试与维护;第6章详细介绍了面向对象的软件工程;第7章介绍了软件工程中涉及的管理方面的相关内容,包括项目计划、软件资源管理、进度管理、人员管理、风险管理等内容。

每章开头的“本章目标”概述了该章的主要内容;每章基本按照基础理论和知识介绍、相关技术和方法介绍、软件工程工具使用介绍和实际软件项目应用介绍的顺序来组织内容;每章小结对该章主要内容进行总结和回顾;在每章的最后给出了一定数量的练习题,帮助读者检验该章的学习效果,加深重点知识的印象。

<<软件工程实用教程>>

内容概要

本书按照典型的软件开发过程来组织内容，旨在培养学生具备软件工程思想以及实际软件开发的能力。

全书共8章，主要内容包括：软件工程的起源，软件工程相关概念，软件工程方法、过程和工具；软件可行性研究及软件需求分析，软件设计，软件编码及实现，软件测试与维护；面向对象的软件工程；软件工程中涉及的管理方面的相关内容，如项目计划、软件资源管理、进度管理、人员管理、风险管理等内容。

本书可作为普通高校计算机相关专业“软件工程”课程的教材，也可供业余计算机和软件开发爱好者参考。

书籍目录

前言教学建议第1章 软件工程概述 1.1 软件概述 1.1.1 软件的概念及特点 1.1.2 软件分类 1.2 软件危机 1.2.1 软件危机的表现与原因 1.2.2 软件危机的启示 1.3 软件工程 1.3.1 软件工程概念 1.3.2 软件工程发展 1.3.3 软件工程目标和原则 1.3.4 软件工程知识体 1.4 软件过程 1.4.1 软件过程概念 1.4.2 软件过程标准 1.4.3 软件生存周期模型 1.5 软件开发方法 1.6 软件工程工具 1.7 软件工程课程学习资源 1.8 “学生档案管理系统”案例介绍 1.9 小结 1.10 练习题第2章 可行性研究及软件需求分析 2.1 可行性研究 2.1.1 项目立项概述 2.1.2 可行性研究内容 2.1.3 可行性研究步骤 2.2 需求分析基本概念 2.2.1 需求分析任务 2.2.2 需求分析步骤 2.2.3 需求管理 2.3 结构化需求分析方法 2.4 结构化分析建模 2.4.1 实体联系图 2.4.2 数据流图 2.4.3 数据字典 2.4.4 状态迁移图 2.5 Visio的功能及使用 方法介绍 2.6 利用Visio绘制“学生档案管理系统”的数据流图 2.7 “学生档案管理系统”软件需求说明书 2.8 小结 2.9 练习题第3章 软件设计 3.1 软件设计的基本概念 3.1.1 软件设计的意义和目标 3.1.2 软件设计原则 3.1.3 软件设计分类 3.2 软件的体系结构 3.2.1 软件体系结构建模 3.2.2 软件体系结构风格 3.2.3 软件质量属性 3.3 软件概要设计 3.3.1 软件概要设计中的重要概念和原则 3.3.2 软件概要设计方法 3.4 软件详细设计 3.4.1 数据设计 3.4.2 界面设计 3.4.3 构件设计 3.4.4 面向数据结构的设计方法 3.5 利用面向数据流的方法设计“学生档案管理系统” 3.6 “学生档案管理系统”软件设计说明书 3.7 小结 3.8 练习题第4章 软件编码及实现 4.1 程序设计语言 4.1.1 程序设计语言的发展与分类 4.1.2 常见程序设计语言介绍 4.1.3 选择程序设计语言的考虑因素 4.2 编码风格 4.3 Visual Studio 4.3.1 Visual Studio界面介绍 4.3.2 Helloworld程序 4.3.3 加法程序 4.3.4 图形界面 4.3.5 调试 4.4 使用Visual Studio实现“学生档案管理系统”用户验证模块 4.4.1 建立数据库和表 4.4.2 编写数据库操作代码 4.4.3 编写页面和逻辑代码 4.5 小结 4.6 练习题第5章 软件测试与维护 5.1 软件测试的基本概念 5.1.1 软件测试原则 5.1.2 软件测试分类 5.1.3 软件测试模型 5.2 软件测试策略 5.2.1 软件测试步骤 5.2.2 软件测试信息流 5.2.3 软件测试文档 5.3 测试用例 5.3.1 测试用例设计方法 5.3.2 测试用例场景 5.4 软件测试方法 5.4.1 等价类划分法 5.4.2 基本路径测试法 5.5 软件维护 5.5.1 软件维护的概念 5.5.2 软件维护过程 5.5.3 软件的可维护性 5.5.4 软件维护的副作用 5.6 使用Visual Studio的Unit Test3功能 5.6.1 新建一个Project 5.6.2 编码 5.6.3 建立Unit Test 5.6.4 进行测试 5.7 “学生档案管理系统”的测试分析报告 5.8 “学生档案管理系统”的使用说明书 5.9 小结 5.10 练习题第6章 面向对象的软件工程 6.1 面向对象概述 6.1.1 面向对象的基本概念 6.1.2 面向对象的实施步骤 6.2 面向对象建模语言 6.2.1 “4+1”视图 6.2.2 UML相关图 6.3 面向对象的分析 6.4 面向对象的设计 6.5 面向对象的实现 6.6 面向对象的测试 6.7 利用Rose工具绘制“学生档案管理系统”的用例图 6.8 利用Rose工具绘制“学生档案管理系统”的顺序图 6.9 利用Rose工具绘制“学生档案管理系统”的类图 6.10 小结 6.11 练习题第7章 软件工程管理 7.1 软件项目管理 7.1.1 软件项目管理概述 7.1.2 项目计划 7.1.3 项目范围管理 7.1.4 项目资源和成本管理 7.1.5 项目时间管理 7.2 软件组织和人员管理 7.3 软件质量保证 7.3.1 软件质量管理 7.3.2 CMM模型 7.4 软件配置管理 7.5 风险管理 7.5.1 软件风险 7.5.2 软件风险管理 7.6 软件文档 7.7 Project的功能及使用方法介绍 7.8 利用Project对“学生档案管理系统”的开发过程进行管理 7.9 小结 7.10 练习题第8章 项目综合实践 8.1 面向对象的分析 8.1.1 收集并整理原始需求 8.1.2 构建并描述用例模型 8.1.3 优化用例模型 8.2 面向对象的设计 8.2.1 确定候选业务对象 8.2.2 确定属性 8.2.3 确定服务 8.2.4 确定关系 8.3 系统实现与测试 8.4 小结 8.5 练习题参考文献

章节摘录

插图：在划分模块时，要注意模块的可分解性、可理解性以及保护性。

可分解性就是指把一个大问题分解为多个子问题的系统化机制。

可理解性是指一个模块可以作为一个独立单元来理解，以便于构造和修改。

保护性是指当一个模块内部出现异常时，它的负面影响应该局限在该模块内部，从而保护其他模块不受影响。

此外，还要注意模块的规模要适中。

模块中所含语句的数量可以用来衡量模块规模的大小。

如果模块的规模过大，那么模块内部的复杂度就会较大，也就加大了日后测试和维护工作的难度。

如果模块的规模过小，那么势必模块的数目会较多，增大了模块之间相互调用关系的复杂度，同时也增大了花费在模块调用上的开销。

虽然并没有统一的标准来规范模块的规模，但是一般认为，程序的行数在50-100范围内比较合适。

采用模块化，不仅降低了问题的复杂度，而且可以实现系统的并行开发，加快了开发进度。

(2) 抽象化抽象是人们认识复杂的客观世界时所使用的一种思维工具。

在客观世界中，一定的事物、现象、状态或过程之间总存在着一些相似性，如果能忽略它们之间非本质性的差异，而把其相似性进行概括或集中，那么这种求同存异的思维方式就可以看做是抽象。

毕竟现实世界中的很多问题是非常复杂的，而人类的思维能力是有限的。

只有运用抽象的思维方法，人们才能有效地解决问题。

通常，在软件项目的开发过程中，人们运用不同层次的抽象。

一个庞大、复杂的系统可以先用一些高级的宏观的概念构造和理解，然后这些概念又可以用一些较微观较细节的概念构造和理解，如此进行，直到最低层次元素。

此外，在软件的生存周期中，从可行性研究到系统实现，每一步的进展也可以看做是一种抽象，这种抽象是对解决方案的抽象层次的一次精化。

在可行性研究阶段，目标系统被看成是一个完整的元素。

在需求分析阶段，人们通常用特定问题环境下的常用术语来描述目标系统不同方面、不同模块的需求。

从概要设计到详细设计的过渡过程中，抽象化的程度也逐渐降低。

而当编码完全实现后，就达到了抽象的最底层。

<<软件工程实用教程>>

编辑推荐

《软件工程实用教程》是结合作者多年从事软件工程课程教学的经验编写而成的，采用理论和实践相结合的方式，按照典型的软件开发过程来组织内容。

为了体现软件工程知识体系的层次，并具有更好的实践指导意义，书中除了介绍软件工程的基本概念、理论、方法和过程外，还十分重视软件工程相关工具的使用，并通过实际案例来讲述软件工程在实际软件项目开发中的应用和体现。

在介绍理论知识的同时，适当选取可操作性和实用性强的实例进行讲解，以加深学生对软件工程思想和方法的理解和掌握。

每章基本按照基础理论和知识介绍、相关技术和方法介绍、软件工程工具使用介绍以及实际软件项目应用介绍的顺序来组织内容，并给出相应的练习题，以加深学生对重点知识的掌握。

最后一章介绍了一个相对完整的项目开发实例，贯穿面向对象的软件工程相关内容，并给出了一些建议练习项目供读者进行实际开发，并同时让读者在这个过程中体会软件工程的基本知识和相关工具的应用。

<<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>