

《面向对象技术与工具》

图书基本信息

书名：《面向对象技术与工具》

13位ISBN编号：9787121171949

10位ISBN编号：7121171945

出版时间：2012-6

出版社：电子工业出版社

页数：404

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《面向对象技术与工具》

内容概要

《软件工程系列规划教材:面向对象技术与工具(第2版)》内容涉及面向对象的思想、方法和技术及两门著名的面向对象程序设计语言——C++语言和Java语言,分为两部分,共12章。第一部分介绍面向对象技术,包括:软件工程概述、软件维护、软件工具与集成化环境、面向对象方法、统一建模语言UML、软件测试;第二部分介绍面向对象程序设计语言,包括:面向对象程序设计语言的核心概念、C++语言实现数据封装、C++语言实现多态性、C++语言实现继承性、Java语言基础、Java语言程序设计。《软件工程系列规划教材:面向对象技术与工具(第2版)》提供配套电子课件。

书籍目录

第1章 软件工程概述

- 1.1 软件工程的产生和发展
 - 1.1.1 软件工程的发展过程
 - 1.1.2 软件危机
 - 1.1.3 软件工程研究的内容
 - 1.2 软件与软件工程过程
 - 1.2.1 软件的概念和特点
 - 1.2.2 软件工程过程
 - 1.3 软件过程模型
 - 1.3.1 瀑布模型
 - 1.3.2 增量模型
 - 1.3.3 螺旋模型
 - 1.3.4 喷泉模型
 - 1.3.5 智能模型
 - 1.4 软件开发方法
 - 1.4.1 结构化开发方法
 - 1.4.2 原型化开发方法
 - 1.4.3 面向对象的软件开发方法
 - 1.5 软件工具与软件开发环境
- 习题1

第2章 软件维护

- 2.1 软件维护的基本概念
 - 2.1.1 软件维护的目的
 - 2.1.2 软件维护的类型
 - 2.1.3 软件维护的特性
 - 2.1.4 软件维护的代价
 - 2.2 软件维护的过程
 - 2.3 软件维护技术
 - 2.4 软件可维护性
 - 2.4.1 软件可维护性的定义
 - 2.4.2 提高可维护性的方法
 - 2.5 逆向工程和再工程
- 习题2

第3章 软件工具与集成化环境

- 3.1 软件工具
 - 3.1.1 软件开发工具
 - 3.1.2 软件维护工具
 - 3.1.3 软件管理与支持工具
- 3.2 集成化CASE环境
 - 3.2.1 概述
 - 3.2.2 集成化的CASE开发环境的要求
 - 3.2.3 集成化的CASE开发环境的体系结构
- 3.3 软件开发工具Rational Rose
 - 3.3.1 Rose工具简介
 - 3.3.2 业务用例图

- 3.3.3 用例图
- 3.3.4 类图
- 3.3.5 协作图与时序图
- 3.3.6 活动图
- 3.3.7 状态图
- 3.3.8 构件图和部署图

习题3

第4章 面向对象方法

- 4.1 面向对象方法概述
 - 4.1.1 什么是面向对象方法
 - 4.1.2 面向对象方法的主要特点
- 4.2 面向对象的基本概念
 - 4.2.1 对象与类
 - 4.2.2 继承
 - 4.2.3 多态性
 - 4.2.4 消息
- 4.3 面向对象的分析
 - 4.3.1 需求分析中的问题
 - 4.3.2 OOA的特点
 - 4.3.3 OOA的基本任务与分析过程
- 4.4 面向对象的设计
 - 4.4.1 OOD的准则
 - 4.4.2 OOD的基本任务
- 4.5 典型的面向对象方法
 - 4.5.1 Booch方法
 - 4.5.2 Coad/Yourdon方法
 - 4.5.3 对象模型技术 (OMT)
 - 4.5.4 OOSE方法

习题4

第5章 统一建模语言 (UML)

- 5.1 UML概述
 - 5.1.1 UML的形成
 - 5.1.2 UML的特点
 - 5.1.3 UML建模及其构成
 - 5.1.4 UML的图形表示
 - 5.1.5 通用模型元素
- 5.2 建立用例模型
 - 5.2.1 需求分析与用例建模
 - 5.2.2 确定执行者
 - 5.2.3 确定用例
 - 5.2.4 建立用例之间的关系
 - 5.2.5 用例建模实例
- 5.3 建立静态模型
 - 5.3.1 类图
 - 5.3.2 包图
- 5.4 建立动态模型
 - 5.4.1 消息
 - 5.4.2 状态图
 - 5.4.3 顺序图

- 5.4.4 合作图
- 5.4.5 活动图
- 5.5 建立实现模型
 - 5.5.1 构件图
 - 5.5.2 配置图
- 5.6 统一过程及其应用
 - 5.6.1 UML与RUP
 - 5.6.2 RUP的特点
 - 5.6.3 RUP的二维开发模型
 - 5.6.4 RUP的迭代开发模式
- 习题5

第6章 软件测试

- 6.1 软件测试概述
 - 6.1.1 软件测试的基本概念
 - 6.1.2 软件测试的特点和基本原则
 - 6.1.3 软件测试过程
 - 6.1.4 静态分析与动态测试
- 6.2 软件测试的策略
 - 6.2.1 单元测试
 - 6.2.2 集成测试
 - 6.2.3 确认测试
 - 6.2.4 系统测试
 - 6.2.5 测试和 测试
 - 6.2.6 综合测试策略
- 6.3 软件调试
 - 6.3.1 软件调试过程
 - 6.3.2 软件调试策略
- 6.4 面向对象的测试
 - 6.4.1 面向对象测试的特点
 - 6.4.2 面向对象测试的类型
 - 6.4.3 分析模型测试
 - 6.4.4 面向对象的测试用例
- 习题6

第7章 面向对象程序设计语言的核心概念

- 7.1 面向对象的目标
- 7.2 面向对象的核心概念
 - 7.2.1 数据封装
 - 7.2.2 继承
 - 7.2.3 多态性
- 7.3 按对象方式思维
- 7.4 面向对象的思想和方法
 - 7.4.1 面向对象是一种认知方法学
 - 7.4.2 面向对象与软件IC
 - 7.4.3 面向对象方法与结构化程序设计方法
 - 7.4.4 对象是抽象数据类型的实现
- 7.5 面向对象的程序设计语言

第8章 C++语言实现数据封装

- 8.1 类的定义
- 8.2 类的成员

- 8.2.1 数据成员
- 8.2.2 成员函数
- 8.2.3 静态成员
- 8.2.4 类外访问成员的方法
- 8.3 C++语言的类
- 8.4 数据封装和信息隐蔽的意义
- 8.5 构造函数
 - 8.5.1 构造函数的作用
 - 8.5.2 构造函数的定义
 - 8.5.3 重载构造函数
- 8.6 复制构造函数
- 8.7 析构函数
- 8.8 对象的创建、释放和初始化
- 8.9 对象和指针
 - 8.9.1 this指针
 - 8.9.2 指向类对象的指针
 - 8.9.3 指向类的成员的指针
- 8.10 友元关系
 - 8.10.1 友元函数
 - 8.10.2 友元类
 - 8.10.3 友元关系的总结
- 8.11 与类和对象相关的问题
 - 8.11.1 类类型作为参数类型
 - 8.11.2 一个类的对象作为另一个类的成员
 - 8.11.3 临时对象

习题8

第9章 C++语言实现多态性

- 9.1 重载运算符
 - 9.1.1 运算符重载的语法形式
 - 9.1.2 重载运算符规则
 - 9.1.3 一元运算符和二元运算符
 - 9.1.4 重载“++”和“ ”的前缀和后缀方式
 - 9.1.5 重载赋值运算符
 - 9.1.6 重载运算符“()”和“[]”
 - 9.1.7 重载输入运算符和输出运算符
 - 9.1.8 指针悬挂问题
- 9.2 C++语言的类型转换
 - 9.2.1 标准类型转换为类类型
 - 9.2.2 类类型转换函数
- 9.3 实例——复数类重载运算符

习题9

第10章 C++语言实现继承性

- 10.1 继承和派生
 - 10.1.1 为什么要使用继承
 - 10.1.2 派生类的声明和继承方式
 - 10.1.3 基类对象的初始化
- 10.2 多继承
 - 10.2.1 多继承的概念
 - 10.2.2 虚基类

- 10.3 继承的意义
 - 10.3.1 模块的观点
 - 10.3.2 类型的观点
- 10.4 虚函数
 - 10.4.1 静态多态性
 - 10.4.2 基类和派生类的指针与对象的关系
 - 10.4.3 虚函数与多态性
- 10.5 纯虚函数和抽象类
- 10.6 虚函数实例——Figure类
- 10.7 类属
 - 10.7.1 无约束类属机制
 - 10.7.2 约束类属机制
- 10.8 模板的概念
 - 10.8.1 函数模板与模板函数
 - 10.8.2 类模板与模板类
- 10.9 实例——一维数组
- 10.10 堆栈、队列的应用

习题10

第11章 Java语言基础

- 11.1 Java语言的发展历程
- 11.2 Java语言的特点
 - 11.2.1 简捷性
 - 11.2.2 面向对象
 - 11.2.3 动态性
 - 11.2.4 安全性
 - 11.2.5 平台无关性和可移植性
 - 11.2.6 高性能
 - 11.2.7 多线程
 - 11.2.8 分布式
 - 11.2.9 健壮性
- 11.3 Java语言的开发工具包
 - 11.3.1 JDK的下载、安装和设置
 - 11.3.2 JDK的简介
- 11.4 Java程序的基本结构
- 11.5 Java程序开发实例
 - 11.5.1 一个简单的Java

.....

第12章 Java语言程序设计

参考文献

章节摘录

版权页：插图：（2）文档管理工具 文档也是软件开发过程中的重要产品。在许多软件的开发过程中，都要花费大量的人力和物力来开发和组织文档，通常，软件开发组织要花20%~30%的工作量来完成软件文档的编写。因此对软件开发过程中产生的文档进行管理和维护对提高软件开发的质量和效率具有重大的意义。文档管理工具用于对软件开发过程中产生的文档进行分析、组织、维护和管理。例如，基于数据流图的需求文档管理工具在对数据流图中的某些成分进行分析时，可以确定该成分的影响范围和被影响范围，以帮助开发成员在对该成分进行修改时，确定其影响范围内的其他成分是否也需要进行相应的变更。而针对源程序文档的文档管理工具可以帮助编码人员确定其中的全局变量或数据结构的作用范围。（3）开发信息库工具 开发信息是指在软件系统的开发过程中，用来维护软件项目开发的相关信息，例如，程序中的对象、模块等内容。开发信息库工具可以用于记录每个对象的开发与修改信息；维护对象与相关信息之间的关系，包括记录对象的开发人员、新版本对象中发生的改动、对象中存在的错误、对该对象进行测试时使用的测试用例、测试结果之间的关系等内容；还可记录用来生成此软件产品的所有开发工具的版本信息、所采用的程序设计语言和应用程序开发接口。（4）逆向工程工具 软件的逆向工程是指对已有的程序进行分析，以获得比源代码更高级的表现形式，是一个设计恢复的过程。逆向工程工具可以帮助软件维护人员从已存在的程序中提取出数据结构、体系结构、程序总体设计等各种有用的软件开发信息。诸如反汇编工具、反编译工具等早期的逆向工程工具，用于将机器代码转换成汇编语言或高级程序语言的代码，以方便开发人员对代码的阅读、理解和修改。现在的逆向工程工具能够分析高级程序语言的源程序，恢复程序的控制结构、流程图、PAD图等更高级的抽象信息，为软件的理解和维护提供方便。（5）再工程工具 软件系统的再工程是指在获得软件设计信息的同时，利用这些信息修改或重构软件系统的工作。根据用户的需求，软件开发人员可利用再工程重新实现已有的软件系统，同时增加新的功能和改进性能。再工程工具可以用来辅助软件开发人员重构一个功能和性能更为完善的软件系统。目前，再工程工具的使用主要集中在代码重构、程序结构重构和数据结构重构等方面。数据结构重构是指通过对数据描述的分析，重新构造出新的数据结构；程序结构重构是指将一个非结构化或结构化程度比较低的源程序改造为一个等价的高度结构化的程序；代码重构是指把一种程序语言书写的程序转成功能等价、但由另一种程序语言书写的或适用于不同硬件平台的程序，例如，将由C语言书写的程序转换为用Java语言书写的功能等价的应用程序。

《面向对象技术与工具》

编辑推荐

《软件工程系列规划教材:面向对象技术与工具(第2版)》可作为高等学校软件工程、计算机等相关专业研究生和高年级本科生的教材,也可作为广大工程技术人员和科研人员的参考书。

《面向对象技术与工具》

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:www.tushu000.com