

<<构件化软件设计与实现>>

图书基本信息

书名：<<构件化软件设计与实现>>

13位ISBN编号：9787302173229

10位ISBN编号：7302173222

出版时间：2008-11

出版时间：清华大学出版社

作者：杨芙清，梅宏 主编，黄罡 等编著

页数：367

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>

<<构件化软件设计与实现>>

前言

软件复用是在软件开发中避免重复劳动的解决方案。

通过软件复用，可以提高软件开发的效率和产品的质量。

近20多年来，面向对象技术、软件构件技术出现并逐步成为主流技术，为软件复用提供了基本的技术支持。

软件复用研究及实践成为热点，被视为解决软件危机、提高软件生产效率和质量的现实可行的途径。

软件复用通常可分为产品复用和过程复用两条途径。

基于构件的复用是产品复用的主要形式，也是当前复用研究及实践的主流。

软件构件技术被视为实现成功复用的关键因素之一。

一、为什么要复用通常情况下，应用软件系统的开发过程包含以下几个阶段：需求分析、设计、编码、测试、维护等。

当每个应用系统的每个部分都是从头开发时，在系统开发过程中就可能存在大量的重复劳动，如：用户需求获取的重复、需求分析和设计的重复、编码的重复、测试的重复和文档工作的重复等。

探讨应用系统的本质，可以发现其中通常包含三类成分：通用基本构件，是特定于计算机系统的构成成分，如基本的数据结构、用户界面元素等，它们可以存在于各种应用系统中。

领域共性构件，是应用系统所属领域的共性构成成分，它们存在于该领域的各个应用系统中。

应用专用构件，是每个应用系统的特有构成成分。

应用系统开发中的重复劳动主要在于前两类构成成分的重复开发。

基于此分类，可以看到，通过凝结共性，通用基本构件和领域共性构件将具有良好的可复用性，由此可以将特定软件的开发聚焦在专用构件的开发及各相关构件的集成组装工作中。

软件复用是在软件开发中避免重复劳动的解决方案，其出发点是应用系统的开发不再采用一切“从零开始”的模式，而是以已有的工作为基础，充分利用过去应用系统开发中积累的知识和经验，如：需求分析结果、设计方案、源代码、测试计划及测试案例等，从而将开发的重点集中于应用的特有构成成分。

<<构件化软件设计与实现>>

内容概要

本书以软件复用为宗旨，以软件体系结构为中心，以中间件为构件框架，系统介绍了构件化软件设计与实现的概念、方法和技术，提供了工程应用范例。

在构件化软件设计方面，突出了软件体系结构的核心地位，重点介绍了如何设计出满足功能与非功能需求、作为构件组装蓝图的软件体系结构。

在构件化软件实现方面，突出了构件框架的主要支撑作用，重点介绍了如何按照软件体系结构、利用构件框架，将可复用构件组装、部署、运营起来。

本书通过理论与实践的结合，使读者不仅学习构件化软件设计与实现的系统化知识，也能有效地掌握实施过程中所需的方法和技术。

<<构件化软件设计与实现>>

作者简介

扬芙清，计算机软件科学家，中国科学院院士，IEEE FELLOW。
现任北京大学信息学部主任、软件工程国家工程研究中心主任、软件与微电子学院理事长。
国务院学位委员会学科评议组召集人，中国软件行业协会副理事长。
《中国科学》、《科学通报》和《电子学报》副主编。

从事系

<<构件化软件设计与实现>>

书籍目录

第1部分 概览	第1章 什么是构件化软件设计与实现	1.1 从编程语言的发展看构件化软件
	1.2 从软件工程学看构件化软件设计与实现	1.3 本书的构件化软件设计与实现方法概貌
	1.4 小结	第2章 构件化软件设计与实现的主要概念
系统结构	2.3 软件构件框架	2.1 软件构件
结构风格	2.4 小结	2.2 软件体系
的集成	第2部分 软件体系结构设计	第3章 软件体系
	3.1 基本概念	3.2 风格的验证
	3.2 典型风格简介	3.4 不同风格
	3.5 小结	第4章 软件体系结构设计
	4.1 体系结构设计方法概述	4.2 体系
	4.2 体系结构设计的主要问题和一般途径	4.3 几种体系结构设计方法介绍
	4.3 几种体系结构设计方法介绍	4.4 面向利用的体系结
	4.4 面向利用的体系结构设计	4.5 小结
	4.5 小结	第5章 软件体系结构评估
	5.1 评估方法的选择	5.2 软件体系结构
	5.2 软件体系结构评估方法的要素	5.3 常见的软件体系结构评估方法
	5.3 常见的软件体系结构评估方法
	第6章 软件体系的结构描述语言	第7章 ABC/ADL与ABCTool
	第7章 ABC/ADL与ABCTool	第3部分 可复用构件开发
	第8章 可复用构件开发基本原则	第9章 面向对象的可复用构件开发
	第9章 面向对象的可复用构件开发	第10章 基于模式的构件开发
	第10章 基于模式的构件开发	第11章 面向EJB/J2EE的
	第11章 面向EJB/J2EE的构件开发	第12章 面向CORBA/CCM的构件开发
	第12章 面向CORBA/CCM的构件开发	第13章 面向COM/.NET的构件开发
	第13章 面向COM/.NET的构件开发	第14章 构件测试
	第14章 构件测试	第4部分 构件组装、部署与运行
	第4部分 构件组装、部署与运行	第15章 构件组装
	第15章 构件组装	第16章 构件部署
	第16章 构件部署	第17章 构件运行维护和管理
	第17章 构件运行维护和管理	第5部分 实例研究
	第5部分 实例研究	第18章 构件运行支撑平台示例——PKUAS
	第18章 构件运行支撑平台示例——PKUAS	第19章 奥运信息系统体系结构建模案例
	第19章 奥运信息系统体系结构建模案例	第20章 信贷管理系统体系结构建模案例
	第20章 信贷管理系统体系结构建模案例	参考文献

<<构件化软件设计与实现>>

章节摘录

第1章 什么是构件化软件设计与实现要点浏览构件化软件设计与实现与其他软件开发方法有着较明显的差异，为什么会有这些差异以及这些差异的具体内涵，是开始构件化软件设计与实现之前必须了解的。

通过阅读本章，读者可以了解到以下几点。

构件化软件与其他软件模型的本质差异。

构件化软件设计与实现与其他软件开发方法的差异。

本书构件化软件设计与实现的内容概览。

1.1 从编程语言的发展看构件化软件是对客观世界中问题空间与解空间的具体描述，是客观事物的一种反映，是知识的提炼和“固化”。

软件开发可以归结为两项主要活动，即对所要解决的问题及其相关事物的认识和基于这种认识所进行的描述。

人类的认识都需借助他们所熟悉的某种自然语言，将认识的结果描述为计算机所能够理解的软件则依赖于编程语言，自然语言和编程语言之间存在较大的“语言鸿沟”，这种鸿沟直接决定了软件开发的难度和复杂度。

因此，提高编程语言的抽象层次，使其蕴含的软件模型具有更强的表达能力、更符合人类的思维模式，成为软件开发方法发展的主线。

在高级语言出现以前，汇编语言（机器语言）是编程的工具，表达软件模型的基本概念（或语言构造）是指令，表达模型处理逻辑的主要概念（机制）是顺序和转移。

这一抽象层次很低，基本等同于计算机的实际计算模型，程序员需要考虑大量的机器细节，与人类思维模式差距极大。

高级语言的出现，例如FORTRAN语言、Pascal语言、C语言等，使用了变量、标识符、表达式等概念作为语言的基本构造，并使用3种基本控制结构（顺序、分支、循环）来表达软件模型的计算逻辑，因此，软件开发人员可以在一个更高的抽象层次上进行程序设计。

随后出现了一系列开发范型和结构化程序设计技术，实现了模块化的数据抽象和过程抽象，提高了人们表达客观世界的抽象层次。

<<构件化软件设计与实现>>

编辑推荐

《构件化软件设计与实现》为软件复用与软件构件技术丛书之一。

<<构件化软件设计与实现>>

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>