<<Java组件设计>>

图书基本信息

书名：<<Java组件设计>>

13位ISBN编号：9787121081651

10位ISBN编号：7121081652

出版时间：2009-4

出版时间：电子工业出版社

作者：孔德生

页数：308

前言

设计，决定了软件质量！

组件化设计和构建软件系统，是实现快速发布高质量软件产品之梦的最佳实践！

组件设计，凝聚了需求分析、面向对象、设计模式、数据结构、多线程等一系列高阶领域的核心精髓！

Java语言，是互联网上的卓越语言，从诞生的那一天开始，就被很多软件工程师所青睐，不断在世界的每个角落开花结果，应用日益广泛。

Java开源世界，如火如荼地发展，无数的框架、组件如雨后春笋般涌现，极大地推进了Java技术在各个领域的商业应用。

然而，由于设计者的眼界、经验和水平等的限制，Java开源框架、组件参差不齐，优秀者并不多见。

在企业软件开发中，Java往往成了臃肿、笨拙、低效等的代名词。

功能最简单的Java组件也要几十、上百个类文件，使用组件需要引入的Jar文件少则几兆字节，多则几十兆、上百兆字节。

因此，所谓鼓吹的轻量级组件，不过是个噱头而已。

笔者从事企业软件开发十余载，专注平台和组件开发领域，深知组件设计与开发过程的“高处不胜寒”。

因此，在本书中，笔者将自己对组件技术的认识和心得进行总结和提炼，为读者奉上一份Java组件设计领域的精彩技术大餐。

<<Java组件设计>>

内容概要

主要讲解企业应用系统中核心组件的设计原则与最佳实践，澄清设计模式、数据结构、多线程、接口设计等多个高阶技术领域中“流行”的认知误区，通过大量的组件实例分析，为读者精彩地讲解组件设计这一最具技术含量的领域需要考虑的核心问题、设计方案与最佳实践。

《Java组件设计》主要涵盖三部分的内容：第1篇，主要讲解了应用程序的架构、组件的定义和组件核心设计原则。

这些原则，都是在大量的企业软件实践中浓缩提炼的精华；第二部分，对Java语言的高级主题进行了详尽的阐述，作为组件实战的必需必备基础功力；第三部分，对企业应用的核心组件，进行需求分析、设计和实现。

核心组件包括：配置组件、Socket通信组件、日志组件、数据库访问组件、JSON转换器和权限组件。

《Java组件设计》主要定位于软件架构师、设计师、高级开发人员在企业实际应用开发中的参考读物，也适合大专院校相关专业作为教材辅导图书阅读。

<<Java组件设计>>

书籍目录

章节摘录

counter是个整型变量，前面用static关键字修饰后，就变成了一个全局变量，在程序的代码执行之前，这个counter变量就已经存在于内存中了，而且是唯一的实例。

counter静态变量是在StaticExample类中声明的，但实际上counter变量是独立于StaticExample的任何实例的。

也就是说，程序没有创建任何StaticExample实例时，counter已经存在。

程序创建100个StaticExample实例时，couner在内存中仍然是一个，而不是100个。

当程序中创建的StaticExample类的实例都被虚拟机垃圾回收了，counter还存在。

因此静态变量的生命周期，可以认为程序的第一行代码执行之前，就已经在内存中准备好，在程序的最后一行代码执行完毕，静态变量还存在于内存中。

静态变量独立于任何对象，包括声明静态变量的类实例对象。

对简单类型是这样，对复杂类型（对象类型）也是这样。

静态变量的这种特性，经常被用在单实例的类实现中，例子如下： 2.static方法 如果一个方法，仅依赖方法的传入参数、其他static变量，则此方法不依赖于声明方法的类实例，应该声明为static。

表示此方法是类方法，而非实例方法。

<<Java组件设计>>

编辑推荐

将澄清设计模式、数据结构、多线程、接口设计等多个高阶技术领域中“流行”的认知误区，通过大量的组件实例分析，为读者精彩讲解组件设计这一最具技术含量的领域需要考虑的核心问题、设计方案与最佳实践。

版权说明