

<<深入浅出设计模式>>

图书基本信息

书名：<<深入浅出设计模式>>

13位ISBN编号：9787113152659

10位ISBN编号：7113152651

出版时间：2013-1

出版时间：郭峰 中国铁道出版社 (2013-01出版)

作者：郭峰

页数：547

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>

<<深入浅出设计模式>>

内容概要

《深入浅出设计模式》总结了许多系统软件在设计开发过程中的难点，力图将设计模式的实际应用与实现原理有机结合起来，破解软件开发人员在学习设计模式过程不能透彻理解并灵活运用设计模式的难题。

所有章节都是先通过具体的示例讲解为什么需要使用某个设计模式，然后讲解该模式的实现原理，最后再通过详细的示例或对很多开源框架进行分析，加深读者对设计模式的理解。

《深入浅出设计模式》适用于中、高级软件设计和开发人员，尤其是已经学习过设计模式但没有收获的开发人员，同时也可作为高校相关专业师生和社会培训班的教材。

<<深入浅出设计模式>>

书籍目录

第1章如何学习设计模式 1.1软件开发遇到的问题 1.2面向对象的设计原则 1.3设计模式的产生和分类 1.4设计模式学习路线 1.5小结 第2章学习设计模式的工具 2.1建模语言概述 2.2设计模式的工具使用 2.3UML类图 2.4小结 第3章单例模式 (Singleton) 3.1哪里会使用到单例模式 3.2单例模式的实现原理 3.3双检测锁机制的单例模式 3.4单例模式在日志管理中的实际应用 3.5单例模式在数据库连接池管理中的实际应用 3.6小结 第4章简单工厂模式 (SimpleFactory) 4.1哪里会使用到简单工厂模式 4.2简单工厂模式的实现原理 4.3简单工厂模式在翻译器中的实际应用 4.4小结 第5章工厂方法模式 (FactoryMethod) 5.1哪里会使用到工厂方法模式 5.2工厂方法模式的实现原理 5.3简单工厂模式与工厂方法模式比较 5.4工厂方法模式在Spring中的实际应用 5.5小结 第6章抽象工厂模式 (AbstractFactory) 6.1哪里会使用到抽象工厂模式 6.2抽象工厂模式的实现原理 6.3抽象工厂模式在翻译器中的实际应用 6.4小结 第7章原型模式 (Prototype) 7.1哪里会使用到原型模式 7.2原型模式的实现原理 7.3原型模式在Java中的实际应用 7.4小结 第8章创建者模式 (Builder) 8.1哪里会使用到创建者模式 8.2创建者模式的实现原理 8.3创建者模式在薪酬模块中的实际应用 8.4小结 第9章适配器模式 (Adapter) 9.1哪里会使用到适配器模式 9.2适配器模式的实现原理 9.3在模块的接口间使用适配器模式 9.4适配器模式在Spring中的实际应用 9.5适配器模式在JUnit中的实际应用 9.6小结 第10章门面模式 (Facade) 10.1哪里会使用到门面模式 10.2门面模式的实现原理 10.3门面模式在Spring JDBC中的实际应用 10.4门面模式在Hibernate中的实际应用 10.5小结 第11章代理模式 (Proxy) 11.1哪里会使用到代理模式 11.2代理模式的实现原理 11.3动态代理的实现 11.4面向方面的程序编程 11.5代理模式在Struts 2中的实际应用 11.6小结 第12章合成模式 (Composite) 12.1哪里会使用到合成模式 12.2合成模式的实现原理 12.3合成模式在JUnit中的实际应用 12.4合成模式在薪酬系统中的实际应用 12.5小结 第13章享元模式 (Flyweight) 13.1哪里会使用到享元模式 13.2享元模式的实现原理 13.3采用单例模式和享元模式来实现数据库连接池 13.4小结 第14章装饰模式 (Decorator) 14.1哪里会使用到装饰模式 14.2装饰模式的实现原理 14.3装饰模式在Java中的实际应用 14.4小结 第15章桥模式 (Bridge) 15.1哪里会使用到桥模式 15.2桥模式的实现原理 15.3桥模式在网上商城系统的实际应用 15.4小结 第16章策略模式 (Strategy) 16.1哪里会使用到策略模式 16.2策略模式的实现原理 16.3策略模式在Spring中的实际应用 16.4小结 第17章迭代器模式 (Iterator) 17.1哪里会使用到迭代器模式 17.2迭代器模式的实现原理 17.3迭代器模式在Java中的具体实现原理 17.4迭代器模式在公交售票系统的使用 17.5小结 第18章模板方法模式 (TemplateMethod) 第19章中介者模式 (Mediator) 第20章访问者模式 (Visitor) 第21章职责链模式 (ChainofResponsibility) 第22章状态模式 (State) 第23章解释器模式 (Interpreter) 第24章观察者模式 (Observer) 第25章命令模式 (Command) 第26章备忘录模式 (Memento) 第27章综合应用设计模式实现MVC框架

<<深入浅出设计模式>>

章节摘录

版权页：插图：27.5 实现持久层功能 在Java开发的系统中，JDBC是程序员和数据库打交道的主要途径，提供了完备的数据库操作方法接口。

但考虑到规范的适用性，JDBC只提供了最直接的数据库操作规范，对数据库资源管理。

在java.sql包中的类，是用于Java应用程序连接数据库的标准方法。

进行Web开发，必然离不开对数据库的访问，因此，需要Java应用程序与各种不同数据库之间进行会话，而JDBC正是作为此种用途的机制。

下面对JDBC的几个常见接口进行介绍：（1）Connection接口代表与数据库的连接。

连接过程包括所执行的SQL语句和在该连接上所返回的结果。

一个应用程序可与单个数据库有一个或多个连接，或者可与许多数据库有连接。

在与数据库建立连接之前，首先要加载对应数据库的驱动程序，加载对应数据库的驱动程序后，即可与数据库建立连接了。

与数据库建立连接的标准方法是调用DriverManager.getConnection方法。

（2）JDBC URL提供了一种标识数据库的方法，可以使相应的驱动程序能识别该数据库并与之建立连接。

由于JDBC URL要与各种不同的驱动程序一起使用，因此这些约定应非常灵活。

JDBC URL可指向逻辑主机或数据库名，而这种逻辑主机或数据库名将由网络命名系统动态地转换为实际的名称。

这可以使系统管理员不必将特定主机声明为JDBC名称的一部分。

（3）DriverManager类是JDBC的管理层，作用于用户和驱动程序之间。

它跟踪可用的驱动程序，并在数据库和相应驱动程序之间建立连接。

另外，DriverManager类也处理诸如驱动程序登录时间限制及登录和跟踪消息的显示等事务。

（4）Statement对象用于将SQL语句发送到数据库中。

它提供了3个类：Statement、PreparedStatement和CallableStatement.用于向数据库发送SQL语句。

Connection接口中的3个方法可用于创建这些类的实例。

下面列出这些类及其创建方法：Statement对象：用于执行不带参数的简单SQL语句，提供了执行语句和获取结果的基本方法，由Connection的方法createStatement所创建。

PreparedStatement对象：用于执行带或不带m参数的预编译SQL语句，由Connection的方法prepareStatement所创建。

PreparedStatement的实例扩展了Statement，因此它们都包括了Statement的方法。

PreparedStatement对象有可能比Statement对象的效率更高，因为它已被预编译过并存放在那以供将来使用。

<<深入浅出设计模式>>

编辑推荐

《深入浅出设计模式》适用于中、高级软件设计和开发人员，尤其是已经学习过设计模式但没有收获的开发人员，同时也可作为高校相关专业师生和社会培训班的教材。

<<深入浅出设计模式>>

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>