

基于MVC模式的Struts框架在财务系统中的应用 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/141/2021\\_2022\\_\\_E5\\_9F\\_BA\\_E4\\_BA\\_8EMVC\\_E6\\_c29\\_141335.htm](https://www.100test.com/kao_ti2020/141/2021_2022__E5_9F_BA_E4_BA_8EMVC_E6_c29_141335.htm) 摘要 对于项目开发人员来

说，要加快系统的开发速度、提高系统的可维护性和可扩展性，首要的任务是划分好系统的层次结构，合理地设计表示逻辑、业务处理逻辑和数据逻辑。MVC模式就是一个很好的将表示、控制和业务处理相分离的解决方案，本文则主要介绍了Struts框架下的财务系统的设计和实现。关键词 MVC模式；Struts框架；Web应用 中图分类号 TP391 文献标识码 A 1

引言 软件开发工作中遇到的很多问题都可能在以往开发同类软件的过程中同样出现过并且已经被解决，解决问题时所用到的好的方法、技术经过积累并不断的归纳、总结形成一种固定的解决方案、一种模式，以便在日后的系统开发过程中专门用来解决同类问题。设计模式就是对以往开发经验的总结，它的提出使开发人员可以简单、方便的复用成功的设计方案，而不必每次都做重复的工作。财务系统具有所涉及的业务逻辑复杂、需求灵活，甚至个别业务可能会在系统使用过程中发生变化的特点，因此如何组织应用程序以使其易于维护和升级便成为摆在开发人员面前的首要问题。早些时候的应用程序开发是用过程化语言来完成的，它们将数据库查询语句这样的数据层代码和HTML这样的表示层代码混合在一起，这样一来无论是对业务逻辑的变更还是对界面显示的修改都会涉及到应用程序的整体，使程序维护起来相当困难。

2 Web层的解决方案MVC模式 MVC模式的提出彻底解决了上述问题。MVC是Model（模型）View（视图）Controller（

控制器)的缩写,它适用于大型可扩展的Web应用的开发,它强制性地 将应用程序的输入、处理和输出分开,将其划分为模型、视图和控制器三个核心部分,使它们各司其职,各自完成不同的任务,其中任何一部分的修改都不会影响其它两部分。在MVC模式中,模型封装了应用问题的核心数据、逻辑关系和业务规则,提供了业务逻辑的处理过程。模型一方面被控制器调用,完成问题处理的操作过程,另一方面为视图获取显示数据提供了访问数据的操作。因为模型是与数据格式无关的,因此一个模型可以为多个视图提供数据,这样一个模型一次编写可以被多个视图重用,从而避免了代码的重复编写。视图是MVC模式下用户看到的并与之交互的界面。视图从模型处获得数据,视图的更新由控制器控制。视图不包含任何业务逻辑的处理,它只是作为一种输出数据的方式。MVC模式中,控制器主要起导航的作用,它根据用户的输入调用相应的模型和视图去完成用户的请求。控制器本身不输出任何东西,它接受用户请求并决定调用哪个模型构件去处理,以及由哪个视图来显示模型处理之后返回的数据。MVC的处理过程是这样的:对于每一个用户输入的请求,首先被控制器接收,并决定由哪个模型来进行处理,然后模型通过业务处理逻辑处理用户的请求并返回数据,最后控制器用相应的视图格式化模型返回的数据,并通过显示页面呈现给用户。图1所示为模型、视图、控制器这三个模块各自的功能以及它们之间的相互关系:图1 100Test 下载频道开通,各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)