

软件模块化

2021年9月18日 20:08

软件模块化让一个复杂的软件系统可以通过多人协作或一步一步的实现来完成整个系统的开发

最简单的按照层次划分软件的方法：分为两层

视图层View：包含了界面长什么样子、数据如何输入和输出，是系统的外壳

模型层Model：负责整个系统的逻辑、数据的存储，是整个系统的核心

视图层比较直接，一般使用按照功能的方式去设计

模型层比较复杂，一般使用面向对象的方式来设计

视图层和模型层之间差异较大，可以增加一个中间层——**控制器层Controller**

这样控制器层可以把面向对象的设计转换为面向功能所需要的的API

同时控制器层也负责把数据从视图层转换为逻辑层所需要的对象

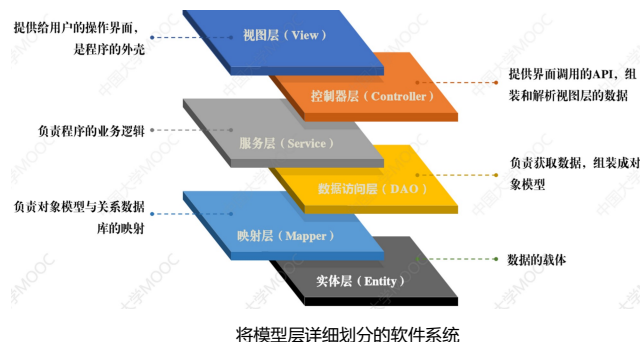
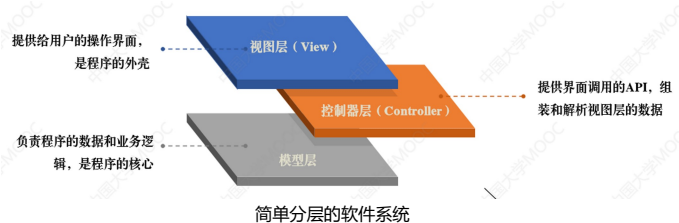
把其中复杂的模型层细分为若干个更小的层次，还可划分出以下层：

服务层Service：负责业务逻辑

数据访问层DAO：负责获取数据组装层对象

映射层Mapper：负责数据对象模型向数据库关系模型转换

实体层Entity：数据的载体



多层次体系结构的优点：

1. 使整个系统的结构更加简单
2. 每一层采用不同的技术来实现，便于不同技能的程序员分工
3. 每一层可以独立测试
4. 代码的变更也可以控制在同层之内

不同层次技术举例：

视图层View：Vue/微信小程序

控制器层Controller：SpringMVC

服务层Service：SpringBean

数据访问层DAO：Redis

映射层Mapper：MyBatis

实体层Entity：不依赖于任何技术，是Plain Old Java Object，即POJO对象



不同层次可以部署在不同的物理服务器上

一般的做法：

把视图层View单独部署在一台Web服务器上，这台服务器上存储静态的网页和图片

把模型层Model部署在Tomcat或Netty所构成的应用服务器上

把数据部署在数据库服务器上

