

MVC

主讲:石小俊

1.体系模型结构

- Model I
 - 纯粹使用jsp
 - 优点
 - 实现简单
 - 开发周期短
 - 对于小幅度的修改，不需要重新编译
 - 缺点
 - 显示逻辑与业务逻辑混乱，可读性太差
 - 代码的复用性较低
 - 不利于大幅度的修改
 - 不利于分工合作
 - 使用jsp+javabean
 - 优点
 - 将显示逻辑与业务逻辑相分离，提高可读性
 - 代码的复用性提高
 - 有利于分工合作
 - 有利于代码的维护
 - 缺点
 - 缺少流程控制
 - 国际化不好实现
 - 异常处理
- Model II
 - 将Servlet纳入到体系模型结构中
 - 以mvc为核心的体系模型结构

2.MCV

是一种软件架构的开发模式

元素	简介	作用
M-Model	数据模型	封装了业务逻辑，对于业务数据进行一定规则的加工
V-View	视图	封装了显示逻辑
C-Contoller	控制器	协调视图与模型

3.MVC运行机制

所有的请求不再直接访问视图与模型

而是统一交给控制器

由控制器分发给合适的视图与模型

如果分发给视图

则选择合适的视图技术来展现

如果分发给模型

由模型实现核心的业务逻辑

实现完成之后也不会直接返回视图

而是将处理结果交给控制器

由控制器选择合适的视图

将结果展现给用户

4.MVC优缺点

- 优点
 - 多个视图共享同一个模型
 - 有利于分工合作
 - 使用者只关心url
 - 方便分工合作
 - 解耦和
 - 简化开发
- 缺点
 - 增加了代码量
 - MVC是一个思想，想要实现需要做详情的需求分析

5.nmms

编号规则

标识符+日期+序列号

- 标识符:由需求定义
 - 例如:
 - 商品---SP
 - 部门---BM
 - 订单---DD
- 日期:年月日
 - 例如:20181101
- 序列号
 - 由多位数字组成
 - 具体多少位由需求决定
 - 假设我们此时对于商品定义其序列号为6位数字
 - 000001-999999
 - 序列号的值都存放于sys_sequence表中
 - id:主键
 - name:标识符
 - value:序列号的值
 - 在sequence表中，每一个标识符对应的数据有且仅有一个
 - 在创建一个新的序列号的时候
 - 需要查询sequence表中是否存在对应标识符的序列号
 - 若不存在
 - 新增一个sequence信息，其name为对应的标识符，value为最小值，000001
 - 若存在
 - 修改对应的数据
 - 若序列号的值达到了最大值999999
 - 则下一个数据为最小值000001
 - 若序列号的值没有达到最大值
 - 则在原有的基础上加1