SpringMVC框架

MVC设计模式

申明是设计模式

设计模式是一套被反复使用多数人知晓的金国分类的代码设计经验的总结

使用设计模式的目的就是为了代码的重用性让代码更容易的被他人理解保证代码的可靠性

设计模式是代码的编写真正的工程化

设计模式是软件工程基石脉络如同大厦的结构一样

Servlet本质上一段Java程序适合处理业务逻辑但是

Servlet不适合向外输出一个Html网页（在servlet中输出网页的通过response获取流 将Html内容一行一行输出到浏览器中这个过程非常繁琐）

HTML：是用于开发网页的一门语言适合作为万股额的输出但是HTML是一个

静态web资源只能输出静态网页无法展示动态的数据

JSP：也可以开发网页并且JSP中可以书写Java代码可以通过Java代码展示动态数据

JSP的出现既解决了Servlet不适合输出网页的问题又解决了HTML无法展示

动态数据的问题

在一些中小型项目中只用JSP进行开发持续了一段事件jsp需要负责的内容

1. 获取请求中携带的数据对请求进行处理
2. 如果在处理请求的过程需要访问数据库在JSP中书写Java代码访问数据库
3. 请求处理的结果仍是用JSP来负责相应
4. 由于JSP中要负责处理很多的内容必然要书写很多Java代码如果JSP中书写了
5. 大量的Java代码会导致JSP内部的Java代码和HTML中混杂在一起会造成JHP
6. 页面结构的很混乱代码的可读性极差带啊吗难以复用会提高以后维护带啊吗的成本

模式2

Servlet+JavaBean+JSP

JSP只负责展示或者相应对请求处理的结果

Servlet+JavaBean：

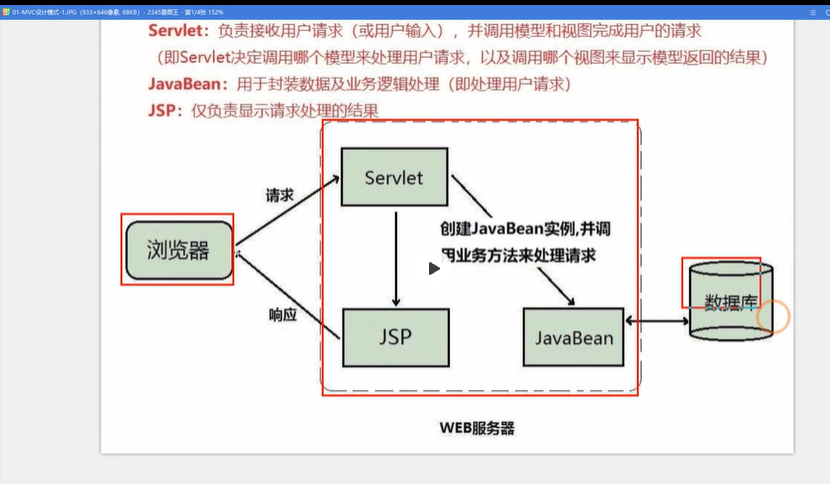
Servlet负责接收用户的请求并且调用模型和试图完成用户的请求

即Servlet决定调用哪个模型来处理用户的请求以及调用哪个试图来显示模型

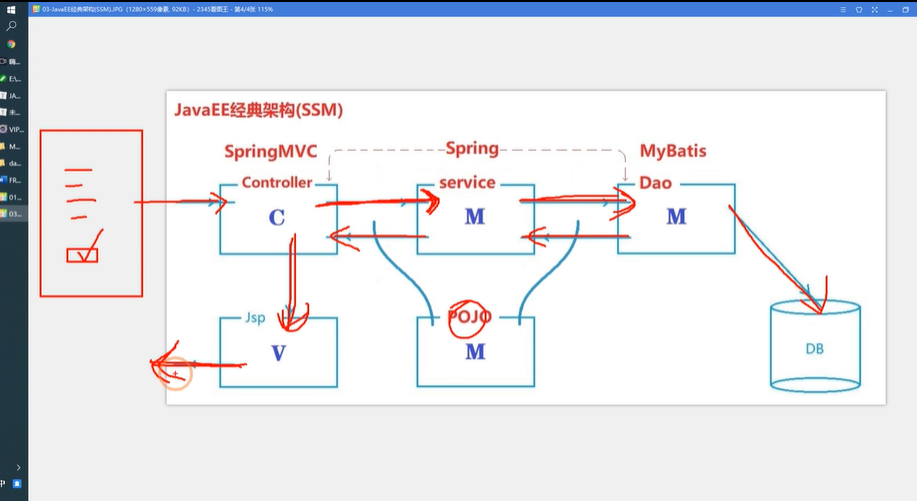
返回的结果

JavaBean用于封装数据以及业务的逻辑处理即处理用户的请求

JSP仅负责显示请求处理的结果



方案3



MVC

设计模式试题中通用的软件变成思想不仅仅适用于Java

在MVC设计模式中认为任何的软件都可以分为三部分

控制程序流转的控制器

封装数据处理数据的模型

负责展示数据的试图

并且在MVC设计思想中认为一个符合MVC设计思想的软件应该

保证上上面这3个都是互相独立的互不干扰的如果一个模块发生了变化

应该尽量不影响其他两个部分这样做的好处就是软件的结构会变得更加的清晰代码的可读性会更强代码也会具有更强的复用性有利于后期 的扩展和维护

SpringMvc简介

Servlet的缺点

通常的情况下一个Servlet类指挥负责处理一个请求若项目中有成千上万个请求

需要处理就需要有成百上千个Servlet类这样会是的项目中的Servlet类个数都会

暴增

在Servlet特别多的时候Web.xml配置量太多不利于团队的开发

当通过客户端提交参数到服务求中通过Servlet进行接收的时候无论数据本身是什么格式在

Servlet中一律按照字符串的格式进行接收后期需要进行类型的转换复杂的类型还需要

进行特殊的处理特别的麻烦

String ParamValue=request.getParameter(String paramName)

Springmvc参数绑定

Servlet具有容器的依赖性必须放在服务器中运行不利于单元测试

SpringMvc：

是Spring框架的一个模块Spring+Springmvc无需中间层的整合

Springmvc是一个基于mvc的web框架

Spring的执行原理

