# DispatcherServlet设计原则

## 对扩展开放，对修改封闭

### 开闭原则的实现

1. 类中所有的变量声明，都在接口中.
2. 使用**模版方法模式**，在父类中对基础行为进行定义，让子类实现模版方法扩展行为。

### 开闭原则的基础

1. 面向对象设计的精髓，是对现实世界中 “行为和契约” 的描述。
2. Java中体现在接口和抽象类的方法声明。
3. 当在越抽象的层次上将这些行为和契约描述清楚后，所设计的系统就是越符合开闭原则的。

## 配置元素的对象化

### 应用背景

1. (概念)在大多数框架都将配置元素集中到外部的 xml 配置文件中，然后在框架的初始化流程中，对这些配置文件进行解析，再变成 java 中的一个个对象供框架使用，这整个过程可以被称为配置元素的对象化。
2. (优点)集中配置，而不是分散到各个类中。

### 实现方式

1. 通过SpringIOC容器完成SpringMVC的配置元素对象化。

### 代码具体实现

1. 将 Spring 容器和 SpringMVC 的初始化过程整合起来。
2. Spring上下文实现类XmlWebApplicationContext中WebApplicationContext 接口，完成配置元素的对象化。‘’

### 2.4 源码分析

## 3. 前端控制器