# 适配器模式改造Servlet

**编写一个Servlet类直接实现Servlet接口有什么缺点？**

我们只需要实现一个service函数，其他函数大部分时候不需要使用。

**适配器模式Adapter**

假设现在有一个接口，其中有多个普通函数（不常用）和一个核心函数（重要函数）。当实际使用的类实现了这个接口后，因为只写了其中一个核心函数的内容，所以会显得代码整体不美观。

解决方案是，首先创建一个适配器类（一个抽象类），实现接口并且重写函数，并将实际会用到的函数设置为抽象函数。然后用实际要用的类继承这个抽象类，重写抽象函数即可。

应用到Servlet中，我们可以写一个标准通用的Servlet类作为适配器，比如GenericServlet类，用它实现Servlet接口，再然后让所有的类都继承GenericServlet类，从而美化和简化实际使用的Servlet类的代码。

**思考：提供了GenericServlet之后，init函数还会执行吗？**

会，会执行GenericServlet中的init函数。（由tomcat服务器调用的）

**思考：GenericServlet类是否需要进一步改造？（模板方法设计模式）**

1. 为init函数参数ServletConfig设计一个成员变量（这个变量在service中可能会使用到），然后在init函数中将config传给成员变量，从而可以调用config。【注意：创建的时候，成员变量config是一个private变量，所以需要用getter获取，从而让子类对象可以获取config变量】
2. 如果子类需要重写GenericServlet的init函数，将会导致父类的init函数不执行，将会进一步导致1中的config无法获取，解决方案如下：【通过设置final关键字，声明不允许子类重写函数，从而保全init被重写的可能性。】
3. 如果子类实在是想要重写init函数，该怎么办？【让父类创建新的init函数重载原来的final函数，让子类重写新的init函数，从而可以在保护原来算法的同时让子类可以重写init函数，注意，为了保证两个init都可以执行，要在原来的final的init函数中调用新的init函数。】

**现成的GenericServlet类**  
在jakarta.servlet包下有一个由Oracle写的现成的GenericServlet类，这是一个现成的GenericServlet，以后所有的类不需要自己创建GenericServlet，而是可以直接继承Oracle给出的GenericServlet类。