

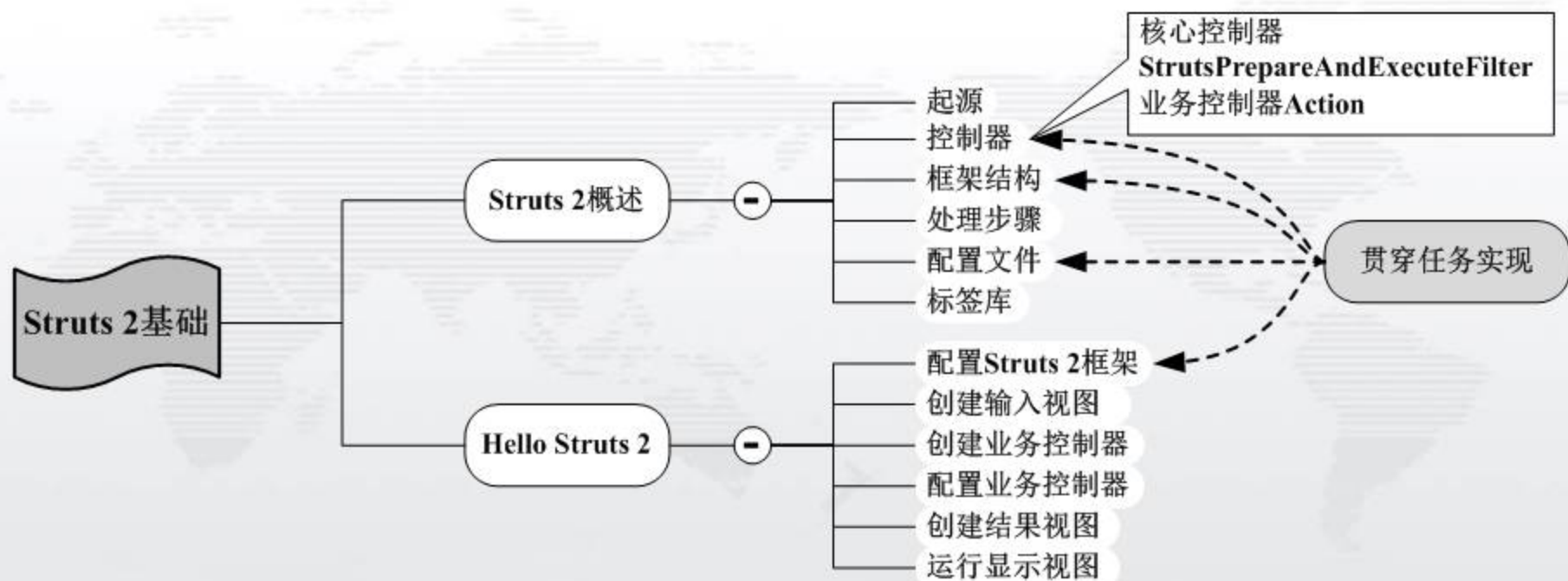
第2章 Struts2基础

- **了解Struts 2的起源**
- **理解Struts 2的框架结构**
- **理解Struts 2的处理步骤**
- **掌握Struts 2的核心控制器**
- **掌握Struts 2的配置文件**
- **掌握第一个Struts2程序的开发**

- 本章任务是了解“GIFT-EMS 礼记”系统的需求及项目框架设计：

【任务2-1】 项目背景介绍及需求分析。

【任务2-2】 项目架构设计。



| 知识点 | 听 | 看 | 抄 | 改 | 写 |
|-----------------|---|---|---|---|---|
| Struts 2起源 | ★ | | | | |
| Struts 2框架结构 | ★ | ★ | | | |
| Struts 2 框架处理步骤 | ★ | ★ | | | |
| Struts 2核心控制器 | ★ | ★ | ★ | ★ | |
| Struts 2配置文件 | ★ | ★ | ★ | ★ | |
| Hello Struts2案例 | ★ | ★ | ★ | ★ | ★ |

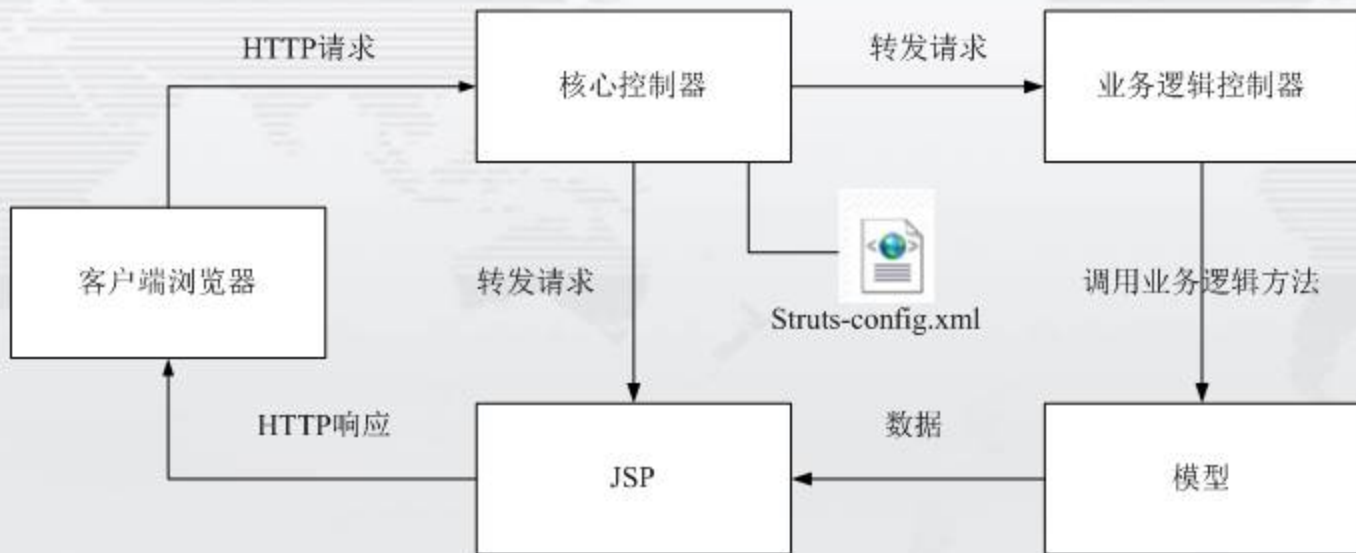
2.1 Struts 2概述

- Struts 2框架是一个MVC开源框架。
- Struts 2是对Struts 1和WebWork两种技术进行兼容、合并的全新框架。
- Struts 2发挥Struts 1和WebWork两种技术的优势，抛弃原来Struts 1的缺点。



2.1.1 Struts 1运行原理

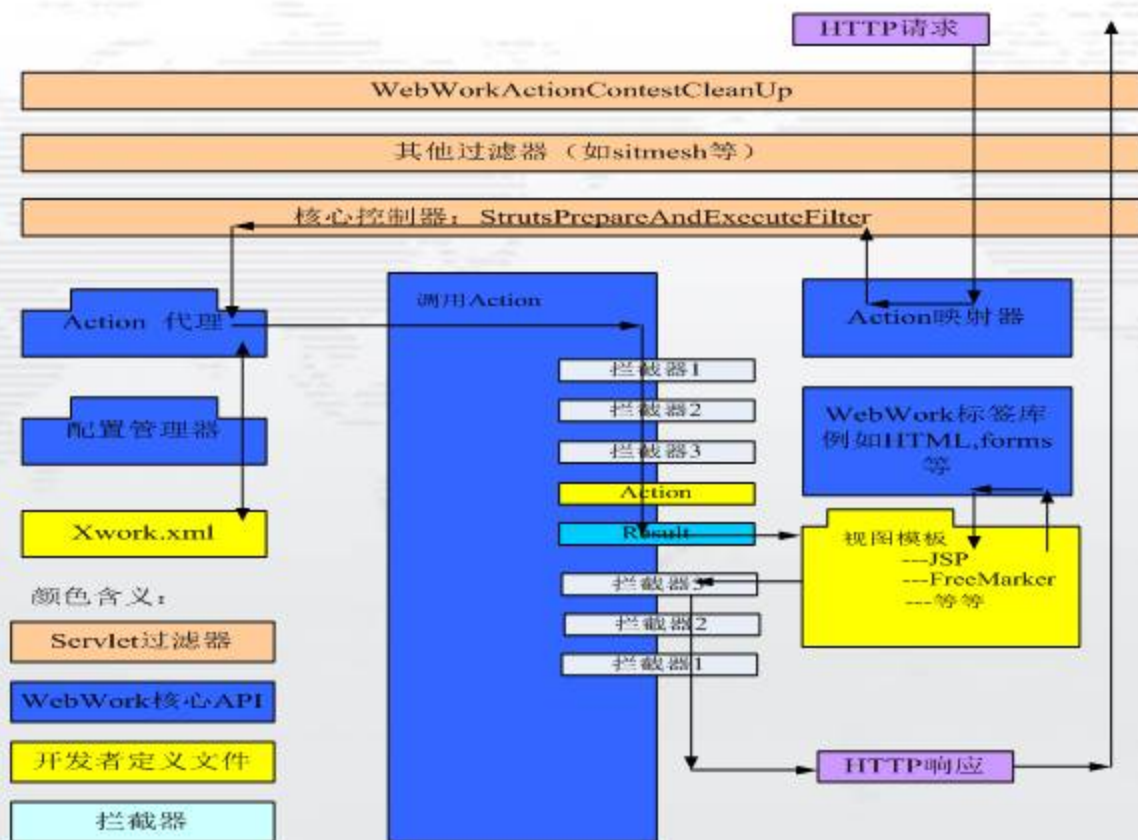
- Struts 1框架的核心是控制器，该控制器由两部分组成：
 - 核心控制器ActionServlet
 - 用户自定义的业务逻辑控制器



- 相对于Struts 1来说，Struts 2的具有如下优势：
 - Struts 2可以自定义显示类型，支持任意表现层技术，有更好的适应性；
 - Struts 2中Action无需跟Servlet API耦合，提高了代码的重用性，也使代码测试更加容易；
 - Struts 2提供了更好的模块化支持和插件机制，增强了框架的可扩展性。

2.1.2 Struts 2框架结构

- Struts 2以WebWork为核心，采用拦截器机制对请求进行处理

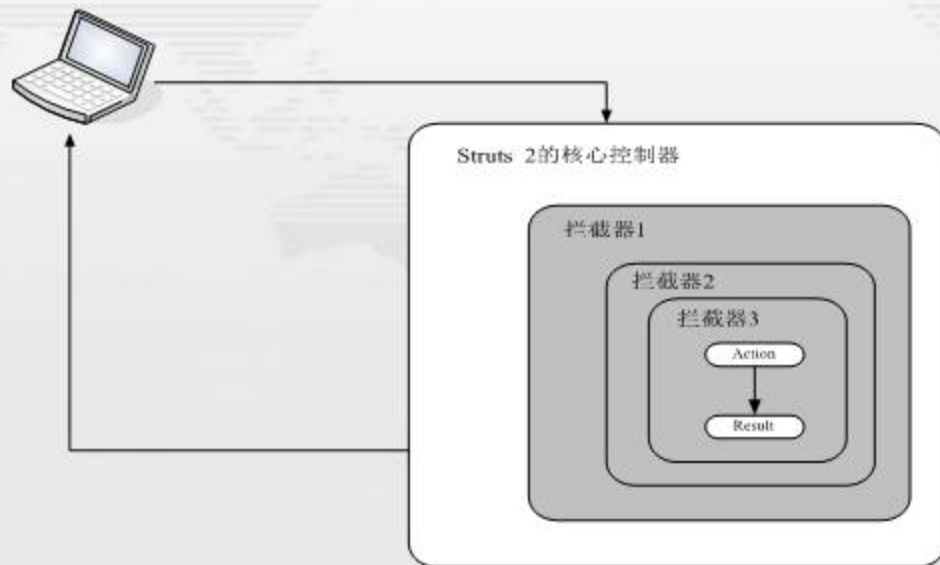


2.1.3 Struts 2处理步骤

● Struts 2的处理流程及步骤如下：

- ① 客户端浏览器发送一个HTTP请求；
- ② Web容器收到请求后，使用ActionContextCleanUp过滤器来清除属性；
- ③ 请求继续经过其他过滤器，然后传递给核心控制器StrutsPrepareAndExecuteFilter；
- ④ 核心控制器确定请求哪个Action，再将控制权委派给ActionProxy代理；
- ⑤ ActionProxy代理调用配置管理器ConfigurationManager从配置文件struts.xml中读取配置信息，然后创建ActionInvocation对象；
- ⑥ ActionInvocation依次调用所用配置拦截器链，然后调用Action，Action执行结果返回结果字符串，ActionInvocation根据该结果字符串查找对应的Result；
- ⑦ Result调用视图模版显示，并在给客户端HTTP响应之前，以相反的顺序执行拦截器链；
- ⑧ 最后，HTTP响应又被返回给核心控制器StrutsPrepareAndExecuteFilter，再依次经过web.xml中配置的过滤器，最终发送到客户端。

- Struts 2的控制器组件是Struts 2框架的核心；
- Struts2的控制器由两部分组成：
 - 核心控制器StrutsPrepareAndExecuteFilter
 - 业务控制器Action



- StrutsPrepareAndExecuteFilter是一个Servlet过滤器；
- 在web.xml中配置Struts 2框架的核心控制器，并使控制器拦截所有请求，示例代码如下：

```
<filter>
  <filter-name>xxxFilter</filter-name>
  <filter-class>
    org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter
  </filter-class>
</filter>
<filter-mapping>
  <filter-name>xxxFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```


2.1.4 业务控制器Action

- 业务控制器Action为一个普通的POJO类，包含一个返回字符串的execute()方法，进行业务处理，示例代码如下：

```
public class xxxAction {  
    //定义属性  
    //属性的getter和setter  
    public String execute() throws Exception {  
        //具体的业务流程  
    }  
}
```

- 实际上，Struts 2中起作用的业务控制器并不是用户定义的动作，而是由Struts 2框架生成的ActionProxy代理，该ActionProxy代理以用户定义的动作为目标。

- **Struts 2的Action类具有如下优势：**

- **Action类完全是一个POJO，提高代码的可重用率；**
- **Action类无须与任何Servlet API耦合，便于测试和应用；**
- **通过Action类中execute()方法的返回值可映射到任何结果视图或其他Action。**

- **Struts 2的配置文件有以下两种：**
 - **struts.xml配置文件，用于配置Action相关信息；**
 - **struts.properties属性文件，用于配置Struts 2的全局属性。**

● 在struts.xml配置Action

```
<struts>
```

```
<!-- Struts 2的Action都必须配置在package里,此处使用默认package -->
```

```
<package name="default" namespace="/" extends="struts-default">
```

```
<!-- 定义一个名为user的Action, 实现类为com.qst.action.LoginAction -->
```

```
<action name="user" class="com.qst.action.LoginAction ">
```

```
<!-- 配置execute()方法返回值与视图资源之间的映射关系 -->
```

```
<result name="success">/main.jsp</result>
```

```
<result name="error">/error.jsp</result>
```

```
</action>
```

```
</package>
```

```
</struts>
```

- **struts.properties属性文件**

- 是Struts 2应用的全局属性文件
- 文件中的内容以“键/值”对的形式存储，即“key = value”的格式

#指定web应用的默认的编码集

struts.i18n.encoding=UTF-8

#当struts.xml修改后是否重新加载该文件，在开发阶段最好打开

struts.configuration.xml.reload=true

#设置浏览器是否缓存静态内容，开发阶段最好关闭

struts.serve.static.browserCache=false。

- **Struts 2标签库的功能如下：**

- **提供了表现层的数据处理**
- **提供了基本的流程控制功能**
- **提供OGNL表达式语言，可以完全替代JSTL标签库**
- **减少了页面代码的书写量，使页面更加简洁**

2.1.6 使用Struts 2标签库

- 在JSP中使用Struts 2标签库，需在页面开头引入

```
<%@taglib prefix="s" uri="/struts-tags"%>
```

- 使用示例：

用户名：<s:property value="userName" />

密码：<s:property value="password" />

姓名：<s:property value="name" />

- 在Struts 2框架下开发Web应用的具体步骤如下：
 - ① 配置Struts 2应用环境；
 - ② 创建输入视图，接收用户输入的数据；
 - ③ 创建业务控制器，并实现其业务处理方法execute()方法；
 - ④ 在struts.xml中配置业务控制器；
 - ⑤ 创建结果视图，显示结果数据；
 - ⑥ 运行Web应用，显示结果页面。

2.2 Hello Struts 2项目效果

输入视图input.jsp

用户信息采集

localhost:8080/chapter02/input.jsp

用户名: 张三

地址: 青岛

电话: 12345678910

邮箱: zhangsan@163.com

提交

提交成功

显示用户信息

localhost:8080/chapter02/user.action

用户名: 张三

地址: 青岛

电话: 12345678910

邮箱: zhangsan@163.com

提交失败

错误页面

localhost:8080/chapter02/user.action

您输入的信息不符合要求, 请重新输入!

- 在Web应用使用Struts 2框架，需要经过以下三步：
 - ① 下载并解压缩Struts 2框架的压缩包；
 - ② 将Struts 2框架的基本类库jar包拷贝到Web应用的“WEB-INF/lib”路径下；
 - ③ 配置Struts 2的核心控制器StrutsPrepareAndExecuteFilter。

- 定义一个表单，将表单提交给“user.action”进行处理

```
<form method="post" action="user.action">
```

```
  用户名: <input type="text" name="userName" /> <br/>
```

```
  地址: <input type="text" name="address" /> <br/>
```

```
  电话: <input type="text" name="telephone" /> <br/>
```

```
  邮箱: <input type="text" name="email" /> <br/>
```

```
  <input type="submit" value="提交" />
```

```
</form>
```

- 创建UserAction类，实现execute()方法

```
public class UserAction {  
    private String userName; // 用户名  
    private String address; // 地址  
    private String telephone; // 电话  
    private String email; // 邮箱  
    /*省略的getter和setter*/  
    public String execute() throws Exception {  
        if (userName.length() > 0 && address.equals("青岛")) {  
            return "success";  
        } else { return "error"; }  
    }  
}
```

● 在struts.xml中配置UserAction

```
<package name="default" namespace="/" extends="struts-default">  
    <!-- 定义一个名为user的Action，实现类为com.qst.chapter02.action.UserAction -->  
    <action name="user" class="com.qst.chapter02.action.UserAction">  
        <!-- 配置execute()方法返回值与视图资源之间的映射关系 -->  
        <result name="success">/result.jsp</result>  
        <result name="error">/error.jsp</result>  
    </action>  
</package>
```

- 在WebContent目录中创建结果视图result.jsp页面

```
<head>
    <title>显示用户信息</title>
</head>
<body>
    用户名: ${param.userName} <br/>
    地址: ${param.address} <br/>
    电话: ${param.telephone} <br/>
    邮箱: ${param.email} <br/>
</body>
```

- 创建错误页面error.jsp

```
<head>  
    <title>错误页面</title>  
</head>  
<body>  
    您输入的信息不符合要求，请重新输入！  
</body>
```

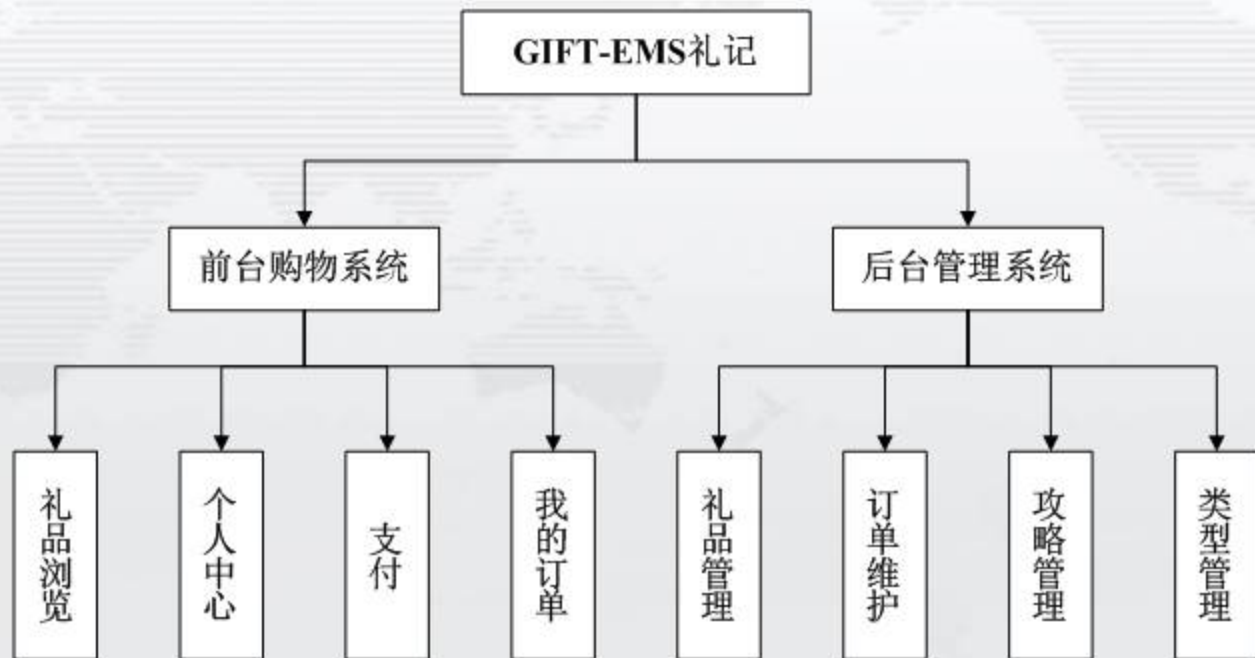
● 项目运行流程：

- ① 输入视图input.jsp接收数据发送请求；
- ② Struts 2的核心控制器拦截到请求转发到对应Action；
- ③ Action返回处理结果，Struts 2根据处理结果返回不同视图。



- **【任务2-1】项目背景介绍及需求分析**
- **【任务2-2】项目架构设计**

● 项目功能模块



● 前台功能模块

| 模块 | 说明 |
|------|--|
| 礼品浏览 | 用户通过礼品列表进入礼品详情界面，同一个礼品有多个款式，不同的款式价格不同 |
| 个人中心 | 包括修改密码、我的订单、安全退出等功能点 |
| 支付 | 用户选购完礼品后，进行购买，生成订单然后进行支付 |
| 我的订单 | 可以查看全部订单、已付款订单、已完成订单、待付款订单等，并且可以对订单进行查看、取消、确认收货等操作 |
| 攻略 | 用户可以浏览并查看攻略详情 |

● 后台功能模块

| 模块 | 说明 |
|------|---|
| 礼品管理 | 进行礼品的添加、修改等维护操作，并最终进行礼品的发布，礼品发布后，前台可以看到新发布的礼品 |
| 订单维护 | 可以对订单进行查看、对礼品进行发货、退款等等操作 |
| 攻略维护 | 后台通过对攻略的增、删、改、查，进行攻略维护 |
| 类型管理 | 对礼品的类型进行增、删、改、查的维护 |

- **“GIFT-EMS礼记”系统采用的技术：**
 - **采用B/S架构，即浏览器/服务器架构；**
 - **使用Struts2作为MVC框架实现模型、视图、控制器的分层；**
 - **数据库使用MySQL，Hibernate框架实现对象的持久化；**
 - **Spring作为基础框架，完成框架的集成。**

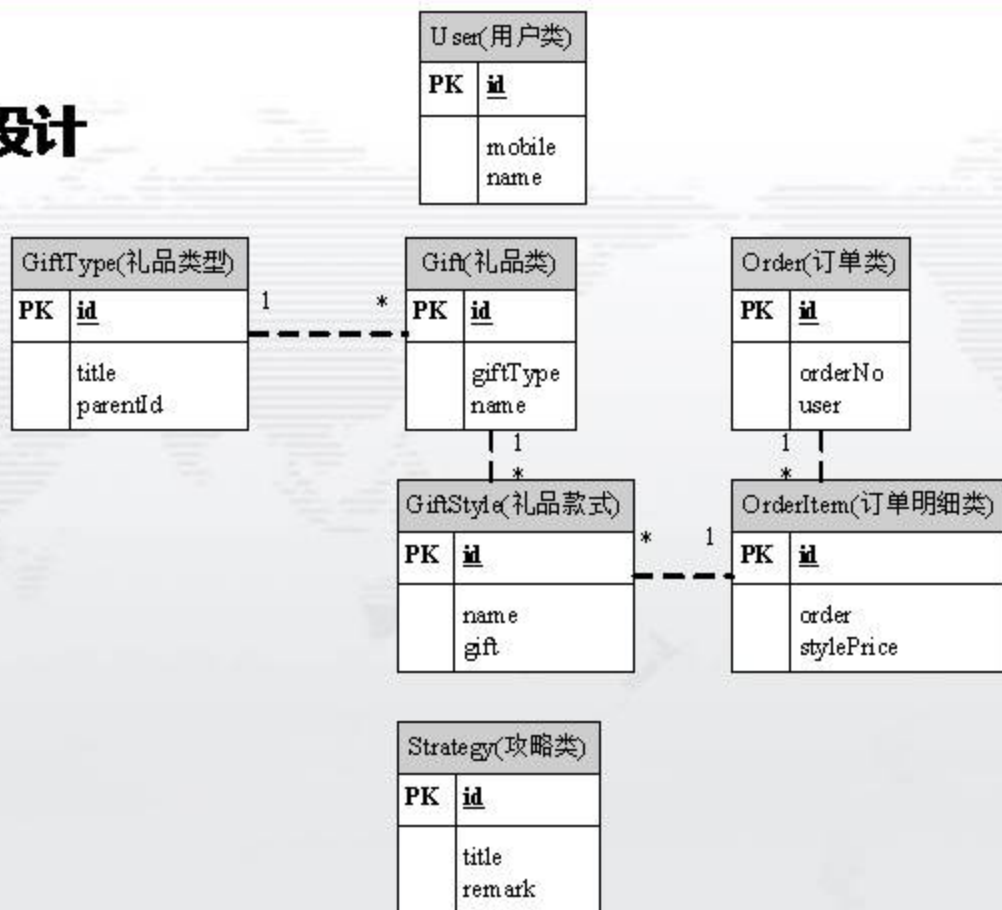
- **“GIFT-EMS礼记”系统表示层采用的技术：**
 - **采用JSP作为表示层技术；**
 - **JSP收集用户的请求信息，提交到控制层特定的控制器；**
 - **控制器把请求结果交与JSP，进行进一步处理，由JSP把结果呈现给用户；**
 - **JSP进行数据的展示或提交时，使用Struts2标签或JSP标签来处理；**
 - **一些交互效果，采用jQuery来实现。**

- **“GIFT-EMS礼记”系统控制层设计如下：**
 - **针对每个业务模块设计控制类Action；**
 - **Action需要在Struts2配置文件中进行配置；**
 - **通过Spring框架的依赖注入实现控制层与业务逻辑层的依赖关系。**

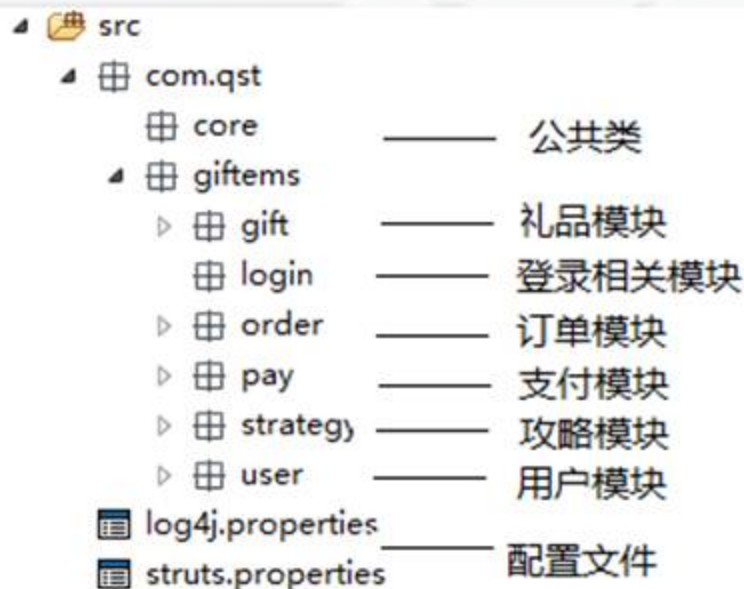
- **“GIFT-EMS礼记”系统业务逻辑层设计如下：**
 - **所有的业务逻辑都封装在业务逻辑层；**
 - **针对每个业务操作在业务逻辑类中都有一个方法与之对应；**
 - **相关联业务操作都封装到同一个业务类中；**
 - **Action会调用对应的业务类中的方法来完成业务逻辑的处理。**

- **“GIFT-EMS礼记”系统数据访问层设计如下：**
 - **数据访问层通常被称为Dao层；**
 - **Dao层采用Hibernate的方式访问数据库；**
 - **Spring通过依赖注入使Struts2和Hibernate进行集成；**
 - **通过Spring IOC容器进行Action对象生命周期的管理；**
 - **系统中的Dao类都继承Spring框架提供的HibernateDaoSupport类。**

● POJO设计



● 项目目录层次结构及功能



- **Struts 2是将Struts 1和WebWork两种技术进行兼容、合并的全新的MVC框架**
- **Struts2大量使用拦截器来处理用户的请求**
- **Struts2的控制器由两部分组成：核心控制器StrutsPrepareAndExecuteFilter和业务控制器Action**
- **在struts.xml文件中定义了Struts 2框架所用的一系列Action**
- **在web.xml中配置StrutsPrepareAndExecuteFilter过滤器后，Web应用就具备Struts 2框架的基本功能支持**

信·未来

信·未来