## J2EE

J2EE是Java2平台企业版（Java 2 Platform,Enterprise Edition）

J2EE核心是一组技术规范与指南，其中所包含的各类组件、服务架构及技术层次，均有共同的标准及规格，让各种依循J2EE架构的不同平台之间，存在良好的兼容性，解决过去企业后端使用的信息产品彼此之间无法兼容，企业内部或外部难以互通的窘境。

J2EE组件和“标准的” Java类的不同点在于：它被装配在一个J2EE应用中，具有固定的格式并遵守J2EE规范，由J2EE服务器对其进行管理。J2EE规范是这样定义**J2EE组件**的：客户端应用程序和applet是运行在客户端的组件；Java Servlet和Java Server Pages (JSP) 是运行在服务器端的Web组件；Enterprise Java Bean (EJB )组件是运行在服务器端的业务组件。

J2EE平台由一整套服务（Services）、应用程序接口（APIs）和协议构成，它对开发基于Web的多层应用提供了功能支持，下面对J2EE中的13种技术规范：

1:JDBC(Java Database Connectivity)

JDBC API为访问不同数据库提供了统一的路径,向ODBC一样,JDBC开发者屏蔽了一些细节问题,另外,JDBC对数据库的访问也具有平台无关性.

3:EJB(Enterprise JavaBean)

J2EE技术之所以赢得广泛重视的原因之一就是EJB。它提供了一个框架来开发和实施分布式商务逻辑,由此很显著的简化了具有可伸缩性和高度复杂的企业级应用程序的开发。EJB规范定义了EJB组件在何时如何与它们的容器进行交互作用.容器负责提供公用的服务，例如目录服务、事务管理、安全性、资源缓冲池以及容错性。但这里值得注意的是，EJB并不是实现J2EE的唯一路径。正是由于J2EE的开放性,使得所有的厂商能够以一种和EJB平行的方式来达到同样的目地。

6:JSP

JSP页面由HTML（标准通用标记语言下的一个应用）代码和嵌入其中的Java代码组成.服务器在页面被客户端所请求以后对这些Java代码进行处理,然后将生成的HTML页面返回给客户端浏览器.

7:Java Servlet

Servlet 是一种小型的Java程序,它扩展了web服务器的功能。作为一种服务器的应用,当被请求时开始执行,这和CGI Perl脚本很相似。Servlet提供的功能**大多和JSP类似**，不过实现的方式不同。**JSP通常是大多数的HTML代码中嵌入少量的Java代码,而servlet全部由java写成并且生成HTML.**

8:XML

XML（标准通用标记语言的子集）是一种可以用来定其它标记语言的语言.它被用来在不同的商务过程中共享数据。XML的发展和java是相互独立的，但是，它和java具有的相同目标是平台独立性。

目前，Java 2平台有3个版本，它们是适用于小型设备和智能卡的Java 2平台Micro版（Java 2 Platform Micro Edition，J2ME）、适用于桌面系统的Java 2平台标准版（Java 2 Platform Standard Edition，J2SE）、适用于创建服务器应用程序和服务的Java 2平台企业版（Java 2 Platform Enterprise Edition，J2EE）

## JavaBean

是一种JAVA语言写成的可重用组件。为写成JavaBean，类必须是具体的和公共的，并且具有无参数的构造器。其实就是实体类吧

它必须满足如下规范：

1）必须有一个零参数的默认构造函数

2）必须有get和set方法，类的字段必须通过get和set 方法来访问。

（get方法无参，set方法有参）

有了JavaBean以后，我们怎么在JSP页面中访问它呢？

方法有2种： 直接访问和JSP标签访问

（不过我们推荐通过标签来访问）

1）直接访问

a) 声明bean

在页面顶部导入JavaBean类： <%@ page import="mytrain.formbean.userBean" %>

在JSP段实例化类： <% userBean user = new userBean(); %>

b) 访问bean

<% user.setXXX(aa); %>

<%=user.getXXX();%>

2）标签访问 （推荐）

a) 声明bean

使用userBean标签：

<jsp:useBean id="user" class="mytrain.formbean.userBean"/>

b) 访问bean

通过setProperty标签来设置：

<jsp:setProperty name="user" property="name" param="mUserName"/>

通过getProperty标签来取得：

<jsp:getProperty name="user" property="name"/>

<http://davidgjy.iteye.com/blog/442749>

## Javaapplet

JavaApplet就是用Java语言编写的小应用程序，可以直接嵌入到网页中，并能够产生特殊的效果。

当用户访问这样的网页时, Applet被下载到用户的计算机上执行，但前提是用户使用的是支持Java的网络l浏览器。由于Applet是在用户的计算机上执行的，因此它的执行速度不受网络带宽或者Modem存取速度的限制。用户可以更好地欣赏网页上Applet产生的多媒体效果。

## SPRING 框架

<http://www.ibm.com/developerworks/cn/java/wa-spring1/>

Spring 是一个开源框架，是为了解决企业应用程序开发复杂性而创建的。框架的主要优势之一就是其分层架构，分层架构允许您选择使用哪一个组件，同时为 J2EE 应用程序开发提供集成的框架。

Spring 框架是一个分层架构，由 7 个定义良好的模块组成。Spring 模块构建在核心容器之上，核心容器定义了创建、配置和管理 bean 的方式，如图 1 所示。

图 1. Spring 框架的 7 个模块



组成 Spring 框架的每个模块（或组件）都可以单独存在，或者与其他一个或多个模块联合实现。

组成 Spring 框架的每个模块（或组件）都可以单独存在，或者与其他一个或多个模块联合实现。每个模块的功能如下：

核心容器：核心容器提供 Spring 框架的基本功能。核心容器的主要组件是 BeanFactory，它是工厂模式的实现。BeanFactory 使用控制反转 （IOC） 模式将应用程序的配置和依赖性规范与实际的应用程序代码分开。

Spring 上下文：Spring 上下文是一个配置文件，向 Spring 框架提供上下文信息。Spring 上下文包括企业服务，例如 JNDI、EJB、电子邮件、国际化、校验和调度功能。

Spring AOP：通过配置管理特性，Spring AOP 模块直接将面向方面的编程功能集成到了 Spring 框架中。所以，可以很容易地使 Spring 框架管理的任何对象支持 AOP。Spring AOP 模块为基于 Spring 的应用程序中的对象提供了事务管理服务。通过使用 Spring AOP，不用依赖 EJB 组件，就可以将声明性事务管理集成到应用程序中。

Spring DAO：JDBC DAO 抽象层提供了有意义的异常层次结构，可用该结构来管理异常处理和不同数据库供应商抛出的错误消息。异常层次结构简化了错误处理，并且极大地降低了需要编写的异常代码数量（例如打开和关闭连接）。Spring DAO 的面向 JDBC 的异常遵从通用的 DAO 异常层次结构。

Spring ORM：Spring 框架插入了若干个 ORM 框架，从而提供了 ORM 的对象关系工具，其中包括 JDO、Hibernate 和 iBatis SQL Map。所有这些都遵从 Spring 的通用事务和 DAO 异常层次结构。

Spring Web 模块：Web 上下文模块建立在应用程序上下文模块之上，为基于 Web 的应用程序提供了上下文。所以，Spring 框架支持与 Jakarta Struts 的集成。Web 模块还简化了处理多部分请求以及将请求参数绑定到域对象的工作。

Spring MVC 框架：MVC 框架是一个全功能的构建 Web 应用程序的 MVC 实现。通过策略接口，MVC 框架变成为高度可配置的，MVC 容纳了大量视图技术，其中包括 JSP、Velocity、Tiles、iText 和 POI。

Spring 框架的功能可以用在任何 J2EE 服务器中，大多数功能也适用于不受管理的环境。Spring 的核心要点是：支持不绑定到特定 J2EE 服务的可重用业务和数据访问对象。毫无疑问，这样的对象可以在不同 J2EE 环境 （Web 或 EJB）、独立应用程序、测试环境之间重用。

<http://www.ibm.com/developerworks/cn/java/wa-spring1/>

<https://www.ibm.com/developerworks/cn/java/wa-spring2/>

<https://www.ibm.com/developerworks/cn/java/wa-spring3/>

我想：实际是用各种XML文件配置实现了松耦合？

## Struts框架

<http://www.ibm.com/developerworks/cn/java/web/struts.html>

## Hibernate框架

在基于MVC设计模式的JAVA WEB应用中，Hibernate可以作为模型层/数据访问层。它通过配置文件(hibernate.properties或hibernate.cfg.xml)和映射文件(\*\*\*.hbm.xml)把JAVA对象或PO(**Persistent Object,持久化对象**)映射到数据库中的数据库，然后通过操作PO，对数据表中的数据进行增，删，改，查等操作。

**Persistent Object**

持久化对象可以是普通的Javabeans,惟一特殊的是它们与（仅一个）Session相关联。JavaBeans在Hibernate中存在三种状态：

1.临时状态(transient):当一个JavaBean对象在内存中孤立存在，不与数据库中的数据有任何关联关系时，那么这个JavaBeans对象就称为临时对象(Transient Object)。

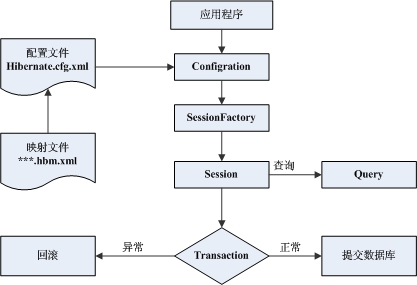
2.持久化状态(persistent):当一个JavaBean对象与一个Session相关联时，就变成持久化对象(Persistent Object)

3.脱管状态(detached):在这个Session被关闭的同时，这个对象也会脱离持久化状态，就变成脱管状态(Detached Object)，可以被应用程序的任何层自由使用，例如可以做与表示层打交道的数据舆对象(Data Transfer Object)。

**Hibernate的运行过程**如下：

A:应用程序先调用Configration类，该类读取Hibernate的配置文件及映射文件中的信息，并用这些信息生成一个SessionFactpry对象。

B:然后从SessionFactory对象生成一个Session对象，并用Session对象生成Transaction对象;可通过Session对象的get(),load(),save(),update(),delete()和saveOrUpdate()等方法对PO进行加载，保存，更新，删除等操作;在查询的情况下，可通过Session对象生成一个Query对象，然后利用Query对象执行查询操作;如果没有异常，Transaction对象将 提交这些操作结果到数据库中。



（1）优点：

a.Hibernate 使用 Java 反射机制 而不是字节码增强程序来实现透明性。

b.Hibernate 的性能非常好，因为它是个轻量级框架。映射的灵活性很出色。

c. 它支持各种关系数据库，从一对一到多对多的各种复杂关系。

（2）缺点：

Hibernate限制您所使用的对象模型。例如，一个持久性类不能映射到多个表。

<http://www.cnblogs.com/eflylab/archive/2007/01/09/615338.html>