**J2EE**

J2EE是一套全然不同于传统应用开发的技术架构，包含许多组件，主要可简化且规范应用系统的开发与部署，进而提高可移植性、安全与再用价值。

**主要特点**

1.J2EE为搭建具有可伸缩性、灵活性、易维护性的商务系统提供了良好的机制：保留现存的IT资产

由于企业必须适应新的商业需求，利用已有的企业信息系统方面的投资，而不是重新制定全盘方案就变得很重要。这样，一个以渐进的（而不是激进的，全盘否定的）方式建立在已有系统之上的[服务器](https://baike.so.com/doc/4487696-4696885.html" \t "_blank)端平台机制是公司所需求的。J2EE架构可以充分利用用户原有的投资，如一些公司使用的BEA Tuxedo、IBM CICS,IBM Encina，、Inprise VisiBroker 以及Netscape Application Server。这之所以成为可能是因为J2EE拥有广泛的业界支持和一些重要的'企业计算'领域供应商的参与。每一个供应商都对现有的客户提供了不用废弃已有投资，进入可移植的J2EE领域的升级途径。由于基于J2EE平台的产品几乎能够在任何操作系统和硬件配置上运行，现有的操作系统和硬件也能被保留使用。

2.高效的开发

J2EE允许公司把一些通用的、很繁琐的[服务端](https://baike.so.com/doc/5988965-6201932.html" \t "_blank)任务交给中间供应商去完成。这样开发人员可以集中精力在如何创建商业逻辑上，相应地缩短了开发时间。高级[中间件](https://baike.so.com/doc/2154861-2280034.html)供应商提供以下这些复杂的中间件服务：

o 状态管理服务 -- 让开发人员写更少的[代码](https://baike.so.com/doc/4469555-4678399.html" \t "_blank)，不用关心如何管理状态，这样能够更快地完成[程序](https://baike.so.com/doc/10037936-10515822.html)开发。

o 持续性服务 -- 让开发人员不用对数据访问逻辑进行编码就能编写[应用程序](https://baike.so.com/doc/3417785-3597266.html" \t "_blank)，能生成更轻巧，与数据库无关的应用程序，这种应用程序更易于开发与维护。

o 分布式共享数据对象CACHE服务 -- 让开发人员编制高性能的系统，极大提高整体部署的伸缩性。

3.支持异构环境

J2EE能够开发部署在异构环境中的可移植[程序](https://baike.so.com/doc/10037936-10515822.html" \t "_blank)。基于J2EE的应用程序不依赖任何特定操作系统、[中间件](https://baike.so.com/doc/2154861-2280034.html" \t "_blank)、硬件。因此设计合理的基于J2EE的程序只需开发一次就可部署到各种平台。这在典型的异构企业计算环境中是十分关键的。J2EE标准也允许客户订购与J2EE兼容的第三方的现成的组件，把他们部署到异构环境中，节省了由自己制订整个方案所需的费用。

4.可伸缩性

企业必须要选择一种[服务器](https://baike.so.com/doc/4487696-4696885.html" \t "_blank)端平台，这种平台应能提供极佳的可伸缩性去满足那些在他们系统上进行商业运作的大批新客户。基于J2EE平台的应用[程序](https://baike.so.com/doc/10037936-10515822.html" \t "_blank)可被部署到各种操作系统上。例如可被部署到高端UNIX与大型机系统，这种系统单机可支持64至256个处理器。（这是NT服务器所望尘莫及的）J2EE领域的供应商提供了更为广泛的负载平衡策略。能消除系统中的瓶颈，允许多台服务器集成部署。这种部署可达数千个处理器，实现可高度伸缩的系统，满足未来商业应用的需要。

5.稳定的可用性

一个[服务器](https://baike.so.com/doc/4487696-4696885.html" \t "_blank)端平台必须能全天候运转以满足公司客户、合作伙伴的需要。因为INTERNET是全球化的、无处不在的，即使在夜间按计划停机也可能造成严重损失。若是意外停机，那会有灾难性后果。J2EE部署到可靠的操作环境中，他们支6.持长期的可用性。一些J2EE部署在WINDOWS环境中，客户也可选择[鲁棒性](https://baike.so.com/doc/5033798-5260234.html" \t "_blank)更好的操作系统如Sun Solaris、IBM OS/390。鲁棒性最好的操作系统可达到99.999%的可用性或每年只需5分钟停机时间。这是实时性很强商业系统理想的选择。

**MVC框架地位**

根据天极网资料显示:基于Web的MVC framework在J2EE的世界内已是空前繁荣，TTS网站上几乎每隔一两个星期就会有新的MVC框架发布，比较好的MVC,老牌的有Struts、Webwork。新兴的MVC 框架有Spring MVC、Tapestry、JSF等。这些大多是著名团队的作品，另外还有一些边缘团队的作品，也相当出色，如Dinamica、[VRaptor](https://baike.so.com/doc/9634017-9979775.html" \t "_blank)等，这些框架都提供了较好的层次分隔能力，在实现良好的MVC 分隔的基础上，通过提供一些现成的辅助类库，同时也促进了生产效率的提高。

如何选择一个好的框架应用在项目中，将会对项目的效率和可重用是至关重要的。

**三大框架及三大框架与MVC之间联系**

hibernate是底层基于jdbc的orm(对象关系映射)持久化框架，即：表与类的映射，字段与属性的映射，记录与对象的映射 数据库模型 也就是Model;  
struts提供的丰富标签是用于显示层View;同时struts也充当了实现Control的功能(不管是Struts1，还是Struts2)，接收参数，视图分发。  
Spring是用于降低层与层间耦合度的，就是所有的类都可以由spring统一创建，用时只需注入即可(IOC)

从功能上来说，hibernate为M，struts2 为V，也可以为C，spring可以为C。  
但大多数项目中，hibernate为M，struts2 为V+C，spring用来帮助管理类以及实现一些业务无关功能，如日志记录等，大多数都是用struts来做C的。  
  
**一、Spring**Spring是一个解决了许多在J2EE开发中常见的问题的强大框架。 Spring提供了管理业务对象的一致方法并且鼓励了注入对接口编程而不是对类编程的良好习惯。Spring的架构基础是基于使用JavaBean属性的Inversion of Control容器。然而，这仅仅是完整图景中的一部分：Spring在使用IoC容器作为构建完关注所有架构层的完整解决方案方面是独一无二的。 Spring提供了唯一的数据访问抽象，包括简单和有效率的JDBC框架，极大的改进了效率并且减少了可能的错误。Spring的数据访问架构还集成了Hibernate和其他O/R mapping解决方案。Spring还提供了唯一的事务管理抽象，它能够在各种底层事务管理技术，例如JTA或者JDBC事务提供一个一致的编程模型。Spring提供了一个用标准Java语言编写的AOP框架，它给POJOs提供了声明式的事务管理和其他企业事务--如果你需要--还能实现你自己的aspects。这个框架足够强大，使得应用程序能够抛开EJB的复杂性，同时享受着和传统EJB相关的关键服务。Spring还提供了可以和IoC容器集成的强大而灵活的MVC Web框架。

**二、STRUCTS**  
Struts是一个基于Sun J2EE平台的MVC框架，主要是采用Servlet和JSP技术来实现的。由于Struts能充分满足应用开发的需求，简单易用，敏捷迅速，在过去的一年中颇受关注。Struts把Servlet、JSP、自定义标签和信息资源(message resources)整合到一个统一的框架中，开发人员利用其进行开发时不用再自己编码实现全套MVC模式，极大的节省了时间，所以说Struts是一个非常不错的应用框架。  
  
**三、Hibernate**  
Hibernate是一个开放源代码的对象关系映射框架，它对JDBC进行了非常轻量级的对象封装，使得Java程序员可以随心所欲的使用对象编程思维来操纵数据库。 Hibernate可以应用在任何使用JDBC的场合，既可以在Java的客户端程序实用，也可以在Servlet/JSP的Web应用中使用，最具革命意义的是，Hibernate可以在应用EJB的J2EE架构中取代CMP，完成数据持久化的重任。Hibernate可以在应用EJB的J2EE架构中取代CMP，完成数据持久化的重任。映射很简单呀,就是页面的映射呀,多做点练习就会了。  
JAVA三大框架用来做WEB应用。 struts主要负责表示层的显示 spring利用它的IOC和AOP来处理控制业务（负责对数据库的操作） hibernate主要是数据的持久化到数据库。在用jsp的servlet做网页开发的时候有个web.xml的映射文件，里面有个mapping的标签就是用来做文件映射的。 当你在浏览器上输入URL地址的时候，文件就会根据你写的名称对应到一个java文件，根据java文件里编写的内容显示在浏览器上，就是一个网页，所以网页名字是随便写的，随便起的名字，不管你起的是.php .jsp .do还是其他的什么都对应这个java文件，这个java文件里的代码进行什么操作就干什么，显示一句话还是连接数据库还是跳转到其他页面等等，这个java文件把数据进行封装起到安全和便于管理的作用。其实这个java文件编译过来是.class的一个字节码文件，没有那么一个类似html嵌入标签和代码的网页文件。他与jsp文件区别就是jsp把代码嵌入到html标签中。 还有servlet或者struts中html表单的action中的提交名称对应一个java文件，struts一般是.do的，都和上面一样是映射。

**mvc的理解**  
MVC是Model—View—Controler的简称。即模型—视图—控制器。MVC是一种设计模式，它强制性的把应用程序的输入、处理和输出分开。  
MVC中的模型、视图、控制器它们分别担负着不同的任务。  
视图: 视图是用户看到并与之交互的界面。视图向用户显示相关的数据，并接受用户的输入。视图不进行任何业务逻辑处理。  
模型: 模型表示业务数据和业务处理。相当于JavaBean。一个模型能为多个视图提供数据。这提高了应用程序的重用性  
控制器: 当用户单击Web页面中的提交按钮时,控制器接受请求并调用相应的模型去处理请求。  
然后根据处理的结果调用相应的视图来显示处理的结果。  
MVC的处理过程：首先控制器接受用户的请求，调用相应的模型来进行业务处理，并返回数据给控制器。控制器调用相应的视图来显示处理的结果。并通过视图呈现给用户。