三层架构（three-tier）与mvc

Three layer 与three-tier的区别

Three layers中的三层是指：

View层： 用于接收用户提交请求的代码在这里编写

Service层：系统的业务逻辑主要在这里编写

Dao层： 直接操作数据库的代码在这里编写

为了降低各层之间的耦合度，在三层架构中使用面向抽象编程，即上层对下层的调用时通过接口实现的。下层对上层的真正服务提供者，是下层接口的实现类。服务标准（接口）是相同的，服务提供者（实现类）可以更换。

而three tier中的三层是指：  
UI（表现层）：主要指与用户交互的界面。用于接收用户输入的数据和显示处理后用户需要的数据。

Bll（业务逻辑层）：ui层和dal层之间的桥梁。实现业务逻辑。业务逻辑具体包含：验证，计算，业务规则等等。

Dal：（数据访问层）：dal是数据库的主要操作系统，实现数据的增删查改等操作，并将操作结果反馈到业务逻辑层bll

MVC

MVC是指：model模型，view视图，controller控件器

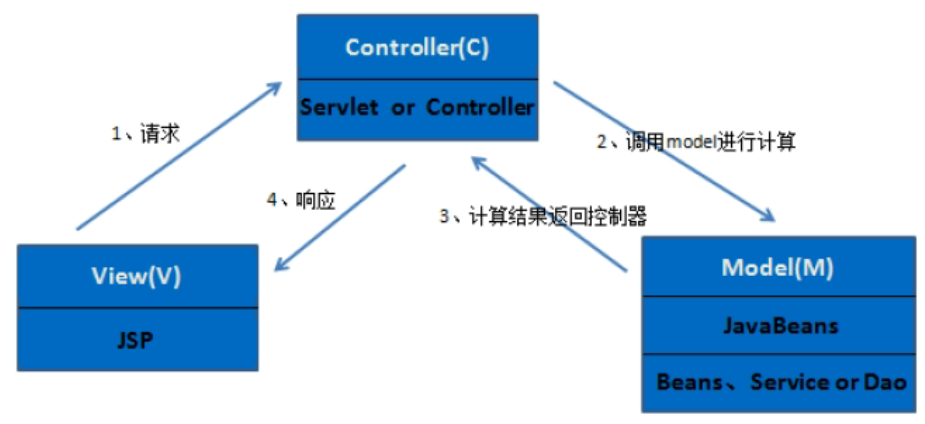
View：视图，为用户提供使用界面，与用户进行直接交互

Model：模型，为承载数据，并对用户提交请求进行计算的模块，一类为数据承载bean,一类成为业务处理bean。所谓数据承载bean是指实体类，专门承载业务数据的，如student, user等。而业务处理bean是指service或者dao对象，专门用于处理用户提交请求。

Controller:控制器，用与将用户请求转发给相应的model进行处理，并处理model的计算结果给用户提供相应响应。

MVC架构程序的工作流程是这样的：

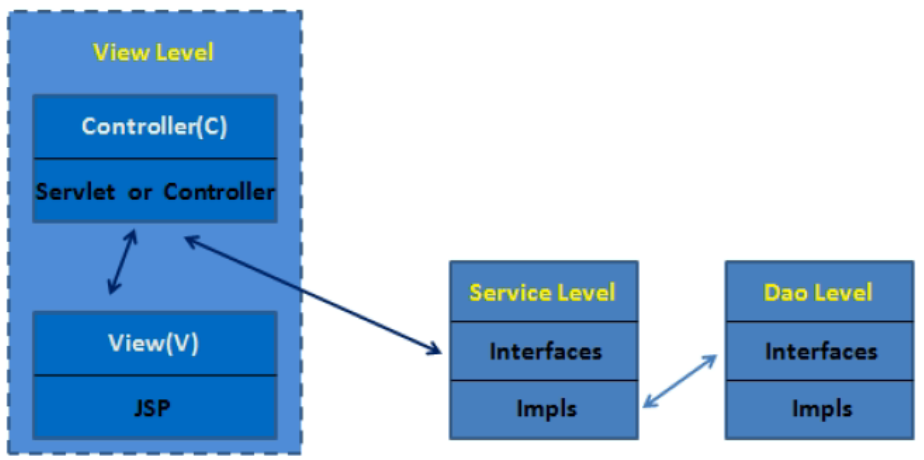
1. 用户通过view页面向服务端提供请求，可以是表单请求，超链接请求，ajax请求等。
2. 服务端controller控制器收到请求后对请求进行解析，找到相应的model对用户请求进行处理
3. Model处理后，将结果再交给controller。
4. Controller在接到处理结果后，根据处理结果找到要作为向客户端发回的相应view页面。页面经过渲染后，再发送给客户端。



MVC与三层架构的关系

MVC与三层架构很相似，但并不一样。

三层架构的view层就是与用户发生直接关系的层，mvc中的v和c就是这样的存在，所以mvc中的v和c均属于三层架构中的view层。同时，mvc中的m（model）包括了数据承载bean和业务处理bean，其中业务处理bean分为service和dao对象，分别对应业务处理逻辑和数据库操作。因此，关系如下图所示：



三层架构的优缺点

优点

1、开发人员可以只关注整个结构中的其中某一层；

2、可以很容易的用新的实现来替换原有层次的实现；

3、可以降低层与层之间的依赖；

4、有利于标准化；

5、利于各层逻辑的复用；

6、结构更加的明确；

7、在后期维护的时候，极大地降低了维护成本和维护时间；

8、避免了表示层直接访问数据访问层，表示层只和业务逻辑层有联系，提高了数据安全性。

9、有利于系统的分散开发，每一个层可以由不同的人员来开发，只要遵循接口标准，利用相同的对象模型实体类就可以了，这样就可以大大提高系统的开发速度。

10、方便系统的移植，如果要把一个C/S的系统变成B/S系统，只要修改三层架构的表示层就可以了。业务逻辑层和数据访问层几乎不用修改就可以轻松的把系统移植到网络上。

11、项目结构更清楚，分工更明确，有利于后期的维护和升级。

缺点

1、降低了系统的性能。这是不言而喻的。如果不采用分层式结构，很多业务可以直接造访数据库，以此获取相应的数据，如今却必须通过中间层来完成。

2、有时会导致级联的修改。这种修改尤其体现在自上而下的方向。如果在表示层中需要增加一个功能，为保证其设计符合分层式结构，可能需要在相应的业务逻辑层和数据访问层中都增加相应的代码。

3、增加了开发成本。