MVC模型与设计模式

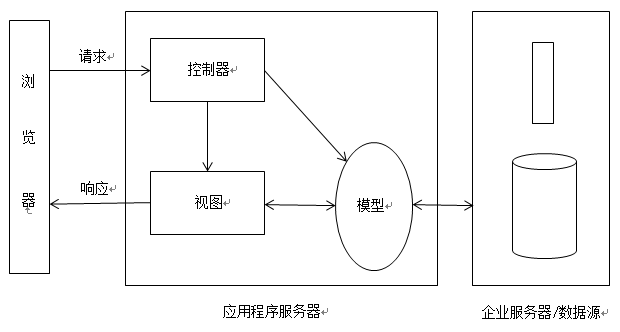
这学期我们既学习了JAVA语言，也学习了23种设计模式，而两者的关系紧密相连，JAVA语言的学习中经常会涉及到设计模式，而设计模式的实现也需要使用JAVA语言。经过一学期的JAVA学习，印象最深刻的应该就是MVC模型。

MVC模型全称为Model-View-Controller，即模型-视图-控制器。Model层一般都是使用Façade设计模式来进行操作，它是在复杂的逻辑上提供简单并且统一的接口，这一层主要就是用来管理应用程序的行为，对数据进行处理；View层主要就是用来获取Controller层的数据处理结果，并且把数据结果显示到浏览器上；Controller层主要是处理一些业务逻辑，根据提出的请求来决定响应的类别，也就是跳转到不同的页面来进行交互处理，Controller层的不同操作会影响到View层和Model层。

MVC在JSP中实现的模型是有两种形式：一是基于Bean的MVC模型，二是基于Struts的MVC模型。而在我们的JAVA学习以及平时做实验中，我们大多采用的是第一种模型，即JSP+Servlet+JavaBean模型。此时，JavaBean就充当Model的角色，主要是用来进行对

从Controller层传过来的数据的操作；Servlet充当Controller的角色，首先是要获取View层前端的数据，然后再调用Model层来对这些前端数据进行处理，最后对数据的不同操作要进行的不同反应，从而跳转到用户所需要的页面；JSP充当View角色，主要用于显示一些经过Model层进行处理过的数据结果。

Model、View、Controller三者的关系如下图所示：



MVC模型主要涉及了观察者以及外观设计模式。MVC中model层涉及了外观模式，model层将一些数据信息进行封装，将数据属性进行私有化，设置为private类型，但是提供了公有public的方法，可以被Controller控制器进行调用，这就使得给外界即Controller提供了一个非常简单的接口，从而实现一些用户请求的功能。除此之外，一个model层可以对应多个View层，因为使用同样的一些数据信息但是可以起到不同的作用，View层显示了整个模型的状态，Controller层根据模型的状态，即用户的请求，可以调用model层所对应的方法来处理这些业务逻辑，之后会自动地更新View层视图业务的状态。

综上所述，MVC模型和设计模式的关系还是很紧密的，可以看出，JAVA语言和设计模式也是不可分离的，这些设计模式思想若能很好的结合JAVA编程语言，则一定能发挥重大作用。

设计模式、XML在信息系统开发中的应用

**一、设计模式在信息系统设计开发中的应用**

这学期我们总共学习了23种设计模式，分别包括了创建型、结构性以及行为型模式。这23种设计模式究其本质其实就是为我们实际当中开发信息系统提供一个指导思想，通过运用这些思想，来降低信息系统内部对象之间的耦合度，提高信息系统的可复用性及易读性,使得系统更加优化、灵活。

由于设计模式是一种将很多功能封装起来，所以如果要修改某一部分不会牵涉到太多的代码，在信息系统开发中，有时候很难获取客户对于系统的完整需求，而且需求也在经常变化，此时采用设计模式就很占优势，只要客户有哪些不满意的功能，那我们就只需要修改实现那一部分功能的代码，而对于外部数据那些都不需要修改。除此之外，很多时候大家会遇到很多类似地问题，这时候，采用设计模式的思想，就可以将原先实现的功能那部分结果直接应用到当前的系统中，这样就可以很快的解决问题，也不用在花时间做无数重复的问题，整体上也能提高系统的效率。

在我们平时做过的信息系统中，其实也有很多涉及到设计模式，比如大二用C#做的医院管理信息系统，一个病人需要跟好几个部门去打交道，就会使得代码变得很复杂，而如果使用设计模式中的外观模式，即设计一个接待员来跟病人接触，而病人的需求就直接告诉接待员，由接待员进行处理挂号、就诊等操作，这样这个接待员就相当于外观模式中的一个高级接口，使得系统变得简单化；如果还想让系统更为简单化，还可以使用单例模式使得系统中只有一个接待员，这样所以病人都只与这一个接待员接触，这样相比每个病人都要与每个部门打交道要容易且简单得多；同时系统中还可以采用builder模式，让复杂对象的构建和表示分离，这样即使是一个非常复杂的功能，也可以让它的构建看起来很简单，而且假如不小心修改了某部分，也很容易能够校准；还可以在此基础上添加工厂方法，这样你想在添加额外的功能，就可以直接创建一个接口、工厂方法及其内部的实现方法就可以了，而不需要对已经写好的代码进行修改，这样就大大提高了系统的可扩展性。

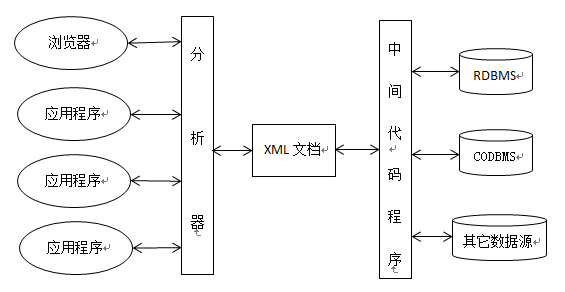
**二、XML在信息系统设计开发中的应用**

这学期我们也对XML进行了学习，对XML也有了一写简单的认识。XML，即可扩展性编程语言，可以随意编写，也可编写有含义的文档，这就与我们以前接触过一些的HTML有一些差距。HTML主要是为了设计界面，本身是没有什么内涵，而XML可以是有含义在里面的，所以在实际的信息系统设计开发中XML也是必不可少的，用来存储信息系统设计开发中的数据。

在我们平时做过的网站中，有很多值得改善的地方，比如上学期电子商务课堂做的购物网站，当HTML文档中需要显示动态数据时，每当数据发生变化时，就需要来修改HTML文档，但是如果我们是使用XML文档进行存储底层的数据，那我们只需要在HTML中进行设计显示的界面，再使用JavaScript来获取这些数据，最后显示到浏览器中即可，这样在修改数据时就不会对HTML进行大量修改，可以提供系统的效率。

而在实际的信息系统设计开发过程中，有更多数据需要涉及到XML文档。在实际开发中，一般都会使用很多种数据库平台，而要在实际中找到完全兼容的数据库很难，此时使用到XML就提供了太多好处。因为XML文档具有非常好的开放性、高度结构化，对于不同数据源之间的交换是非常适合的，通过中间的代理程序来将这些数据源都转换为XML文档，这样这些异构的数据库就有了一个统一的形式了，这样再对XML文档进行处理就很快能实现用户所需的功能了。

XML文档在信息系统开发中的应用模式如下图所示：



**三、总结**

综上所述，设计模式、XML二者与信息系统开发设计的关系都是相辅相成的，尤其是在实际的信息系统开发中，要考虑到现实的很多情况，有了设计模式的思想和XML文档对数据的存储，能够呈现一个很全面的系统。

**控制器**

接受用户请求

调用模型响应用户请求

选择视图显示模型处理后的结果

**视图**

把用户输入的数据传给控制器

显示模型状态

响应模型状态更新

**模型**

封装应用程序状态

响应状态查询

处理业务流程

通知视图更新状态

用户请求

**控制器**

接受用户请求

调用模型响应用户请求

选择视图显示模型处理后的结果

**视图**

把用户输入的数据传给控制器

显示模型状态

响应模型状态更新

**模型**

封装应用程序状态

响应状态查询

处理业务流程

通知视图更新状态

用户请求

选择视图

状态查询

通知更新

响应请求

MVC模式示意图

**应用程序层**

****

主屏幕

联系人

电话

浏览器

.......

**应用程序框架层**

活动管理器

包管理器

窗口管理器

电话管理器

内容管理器

资源管理器

视图系统

本地管理器

通知管理器

XMPP服务

**库**

接口管理器

3D引擎

2D图形引擎

媒体框架

位图及矢量

中间协议

数据存储

浏览器引擎

Lib C函数库

**Android运行环境**

持久层库

Dalvik虚拟机

**Linux内核层**

显示驱动

USB驱动

相机驱动

键盘驱动

蓝牙驱动

WiFi驱动

Flash内存驱动

音频驱动

Binder IPC

驱动

能源管理

**Android系统架构示意图**

1. 移动计算的发展现状

随着无线网络技术的发展，以智能手机为代表的无线移动设备能够随时随地的接入到互联网中，移动计算就是实现移动设备在无线网络中能够进行数据传输和资源共享的一种技术。

而现在移动计算技术在我们的日常生活中已经非常普遍了，当我们在连入无线网来获取网上的一些资源时，移动计算技术就可以将有用、准确的信息及时地响应给需要的用户。这样，智能手机用户就可以在任何能连接无线网络的任何地方使用手机应用程序，因此，移动计算技术在一定程度上给用户带来了极大的便利。

初始化：t=0;NC=0;pij(t)=c; pij(t)=0;将m个

快递员的初始位置均设为快递公司坐标

设置禁忌表索引s=1,并将m个快递员的初始位置加入到禁忌表中

禁忌表已满？

s=s+1;

将m个快递员按照各自计算的转移概率Tij k(t)来选择下一个派送地点，并将该地点加入到禁忌表中

计算所有m个快递员走过的周游长度Lk;更新当前的最优路径

计算pijk ,更新信息素; t=t+n; NC=NC+1;

NC>初始化迭代次数？

输出最优结果

清空禁忌表

Y

N

Y

N