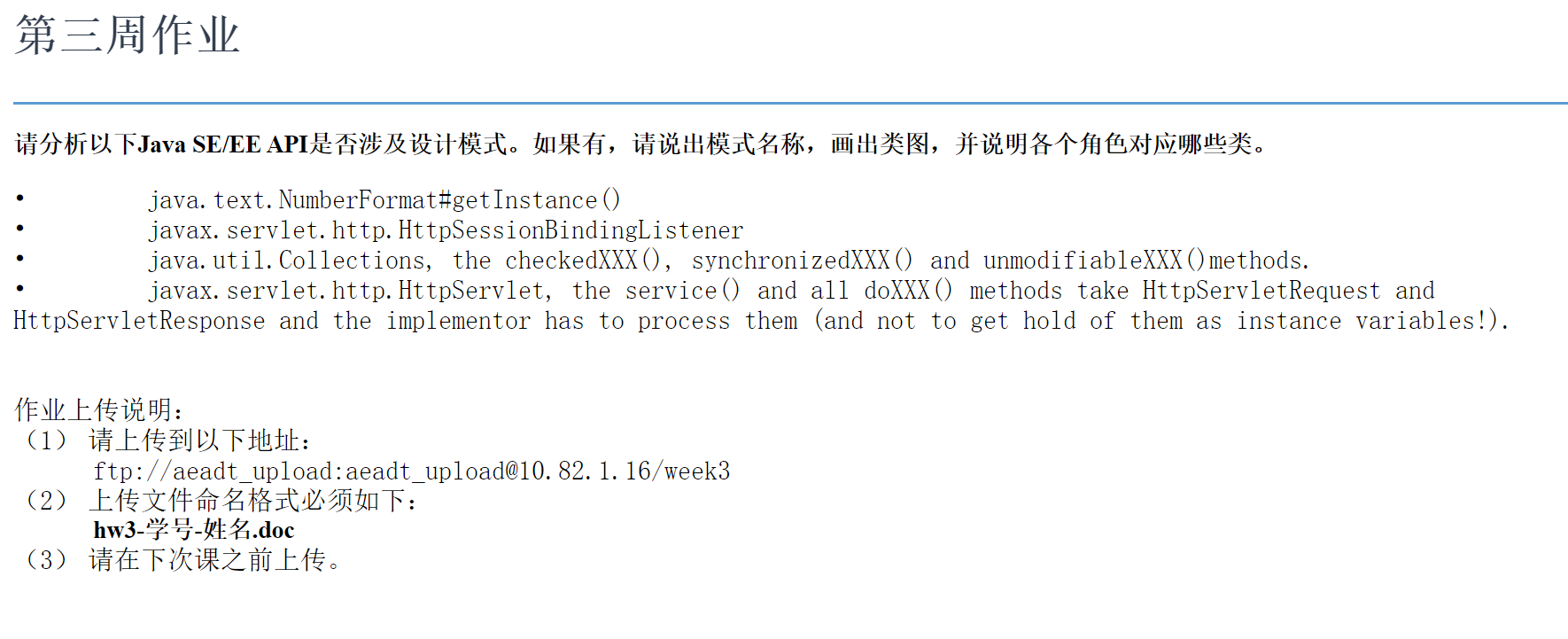
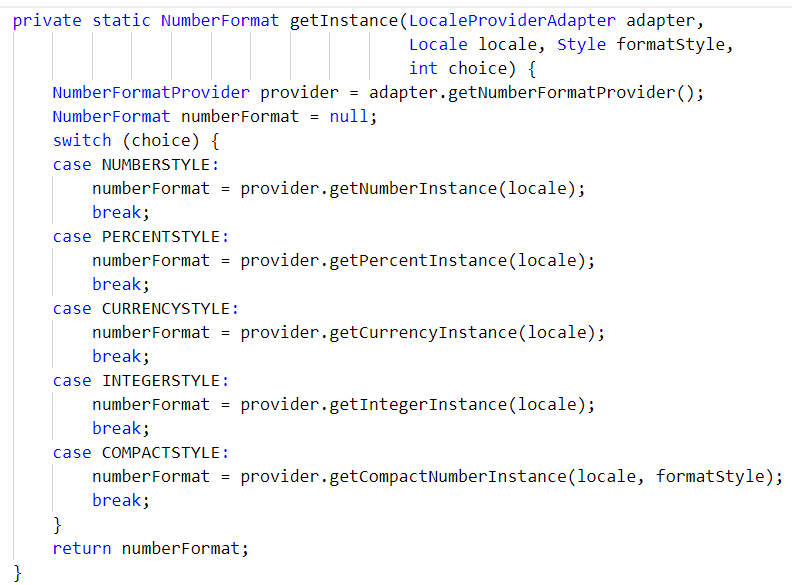
hw3-21951087-胡军.doc

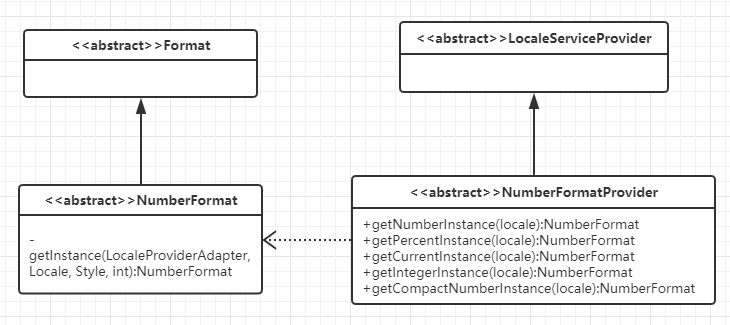


## java.text.NumberFormat#getInstance()

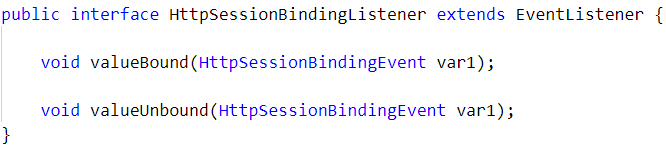


有设计模式，NumberFormat#getInstance属于工厂方法模式。

在getInstance方法中，工厂类NumberFormatProvider根据传入参数类型来制造出不同的NumberFormat产品，以满足客户需求。

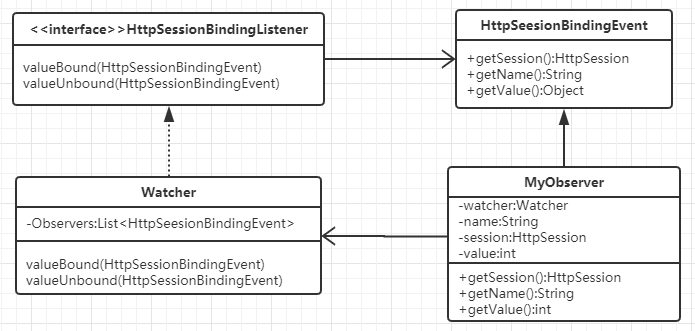


## Javax.servlet.http.HttpSessionBindingListener

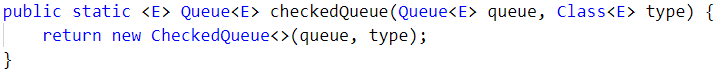


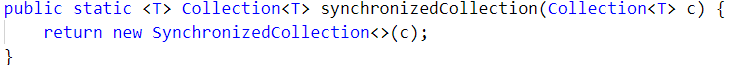
有设计模式，HttpSessionBindingListener使用了观察者模式。

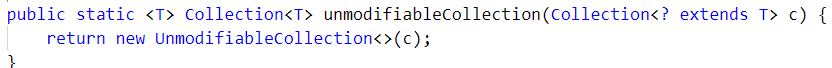
其中，HttpSessionBindingListener作为Subject主题类具有绑定和解绑Observer观察者HttpSessionBindingEvent的能力，另外需要一个Subject继承子类Watcher类来实现两个方法并建立一个观察者列表以便更新时通知观察者。



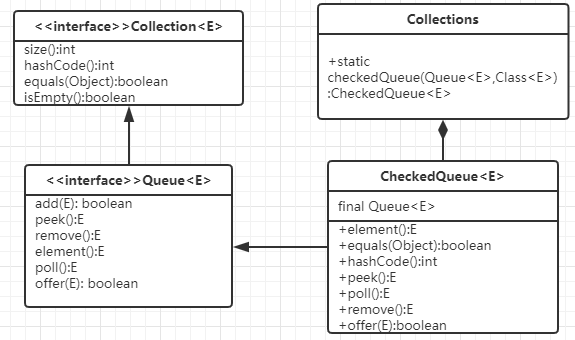
## Java.util.collection



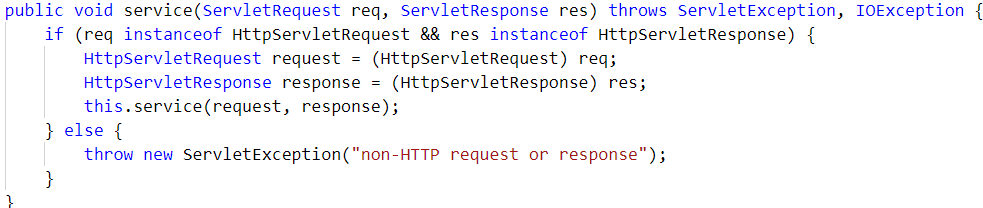


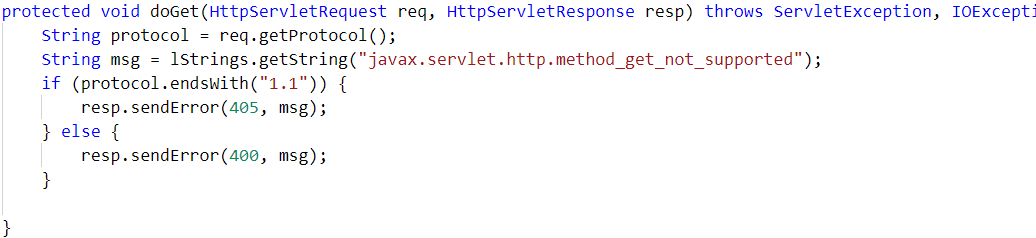


有设计模式和设计原则，Collection类中的各种前缀方法checkedXXX() | synchronizedXXX() | unmodifiableXXX() 使用了装饰模式和开闭原则。由于很多都是相识的，这里举例checkedQueue()方法说明。现在有一个Queue类实现了基本队列功能，假设我们需求要一个能够动态类型检测并可以返回队列视图的新Queue，那么我们应该怎么设计呢？这里就要遵守开闭原则，对扩展开发对修改封闭，所以我们不能直接去改原来的Queue，而是要实现一个新的CheckedQueue，正如上面代码所示。我们可以看到传入参数是Queue，一个普通的队列，然后对这个Queue装饰以后返回一个CheckedQueue类。



## Javax.servlet.http.HttpServlet



有设计模式，HttpServlet用到了策略模式。我们可以看到HttpServlet类有很多doXXX() 的方法。这些方法传入的参数都是一样的，均为HttpServletRequest和HttpServletResponse。每个do方法分别对传入的两个参数改变其状态以达到各自的功能。Service() 同理。

