三、设计模式

1、简单工厂模式

提供一个工厂类Factory用于创建该项目所需的各种类，在该类中提供了一个创建类的工厂方法public object produce(string name)，该方法可以根据所传入参数的不同创建不同的对象。

示例：

Factory fac = new Factory();

Reservation reservation = (Reservation)fac.produce("reservation");

2、观察者模式

观察目标是数据是否更新。

观察者是当前窗体，是用户主界面（目前在用户主界面实现此功能）。

具体表现在：

（1）用户预约信息数据库交互的类ReservationDaoImpl中的插入方法和删除方法，在插入方法的结尾处添加notifyAllObservers(reservation, form)方法，通知观察者更新数据，如果有预约，即在选择完地点时间后，点击确认预约按钮，则需要插入数据库一条新数据，会通知当前的窗体更新listview中的数据，删除同理。

（2）在ReservationDaoImpl中的更新方法中也可加入notifyAllObservers(reservation, form)方法，当取消预约时更新当前状态。

示例：预约前.png>>预约后.png（当点击确认预约按钮后）