**食堂智能餐饮系统**

小组成员：2019312110042黄宸志 2019312110122范纯恺

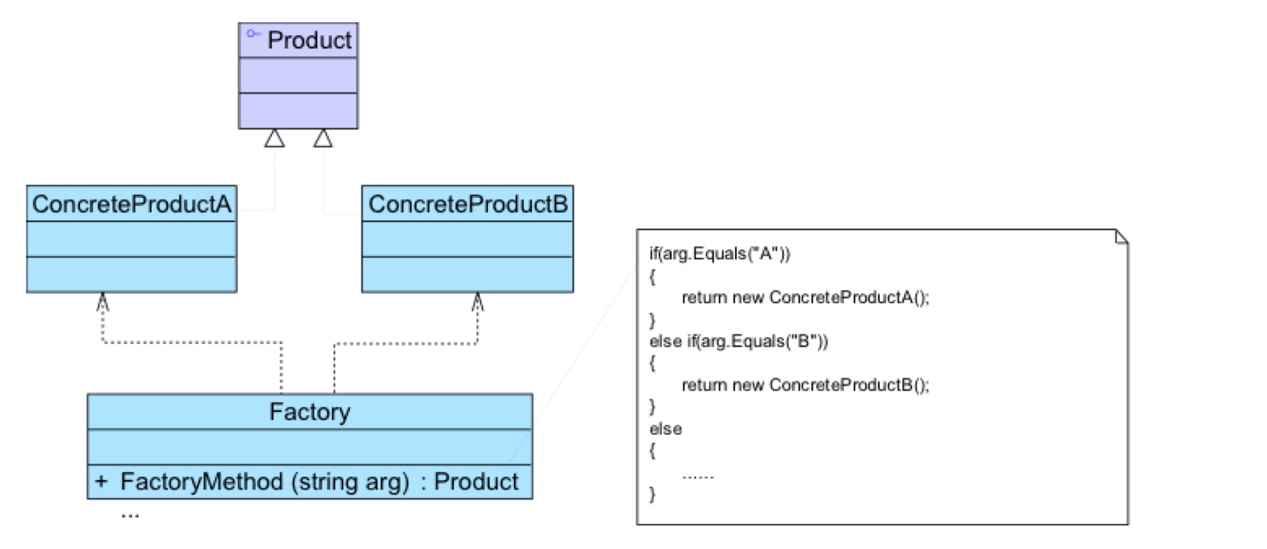
**一、设计模式介绍**

**1.简单工厂模式**

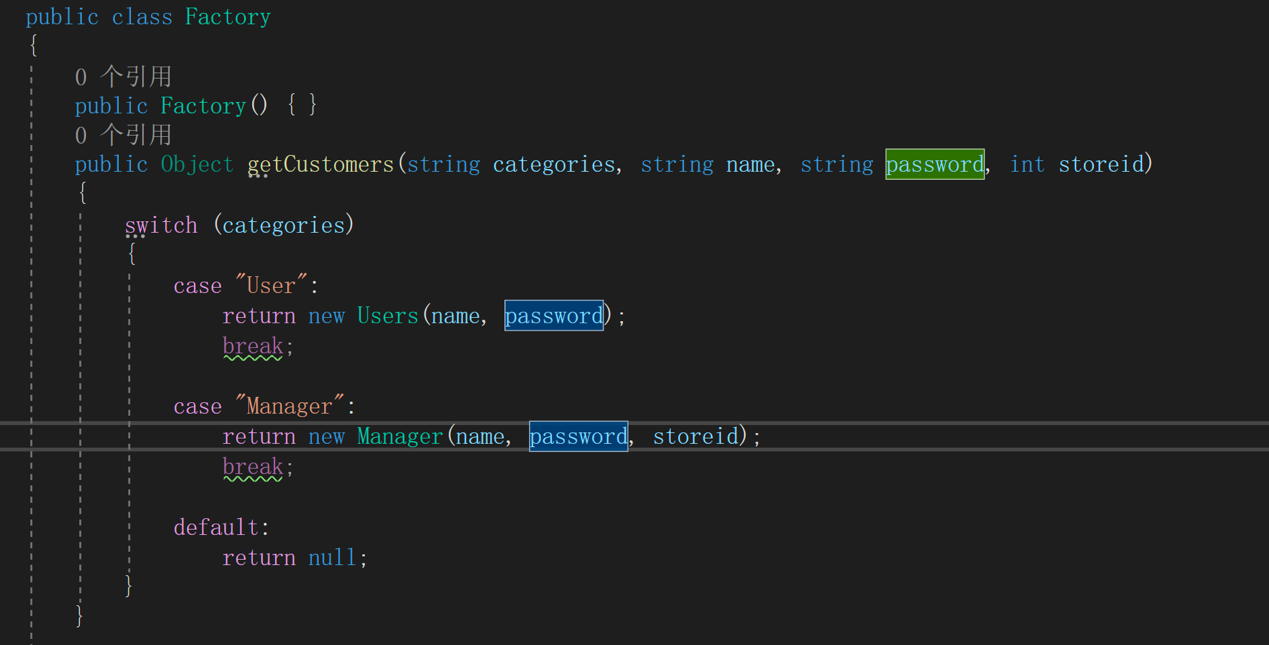
简单工厂模式定义一个工厂类，它可以根据不同参数的不同返回不同类的实例，被创建的实例通常都具有共同的父类。

简单工厂模式的优点：

* 实现了对象创建和使用的分离；
* 客户端无需知道所创建的具体产品类的类名；
* 可以在不修改任何客户端代码的情况下更换和增加新的具体产品类；



我们在餐饮系统中也运用了简单工厂模式，通过客户端传入的参数我们可以更加方便的生成我们想要的类型，例如用户，食物，商店等等的类，不需要我们自己去判断，同时也更方便开发者去修改和增加代码。

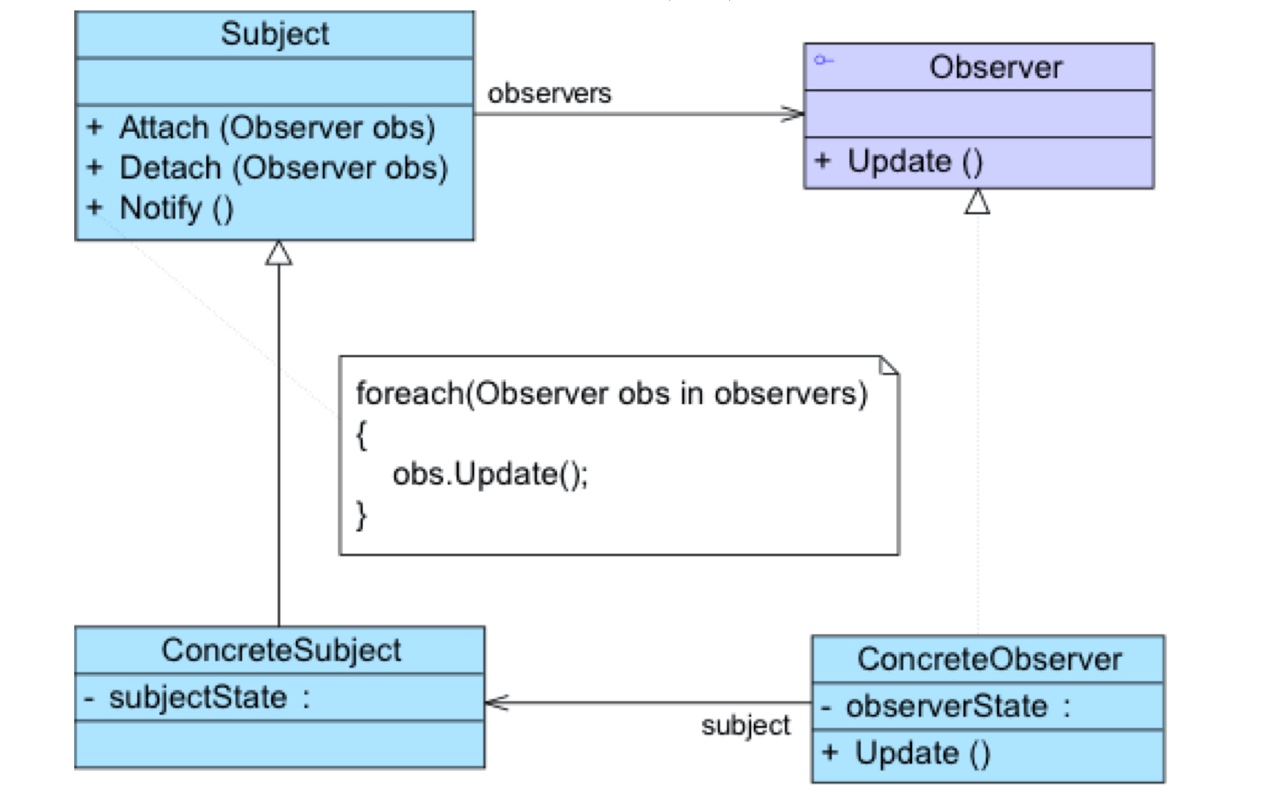


**2.观察者模式**

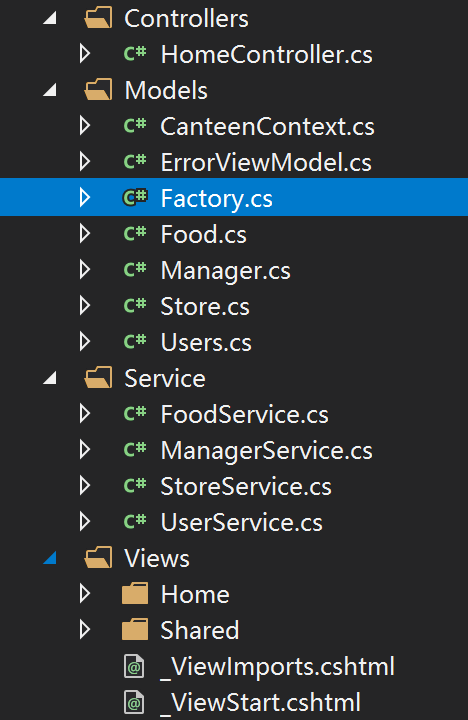
观察者模式定义对象之间的一种一对多依赖关系，使得每当一个对象状态发生改变时，其相关依赖对象都得到通知并被自动更新。

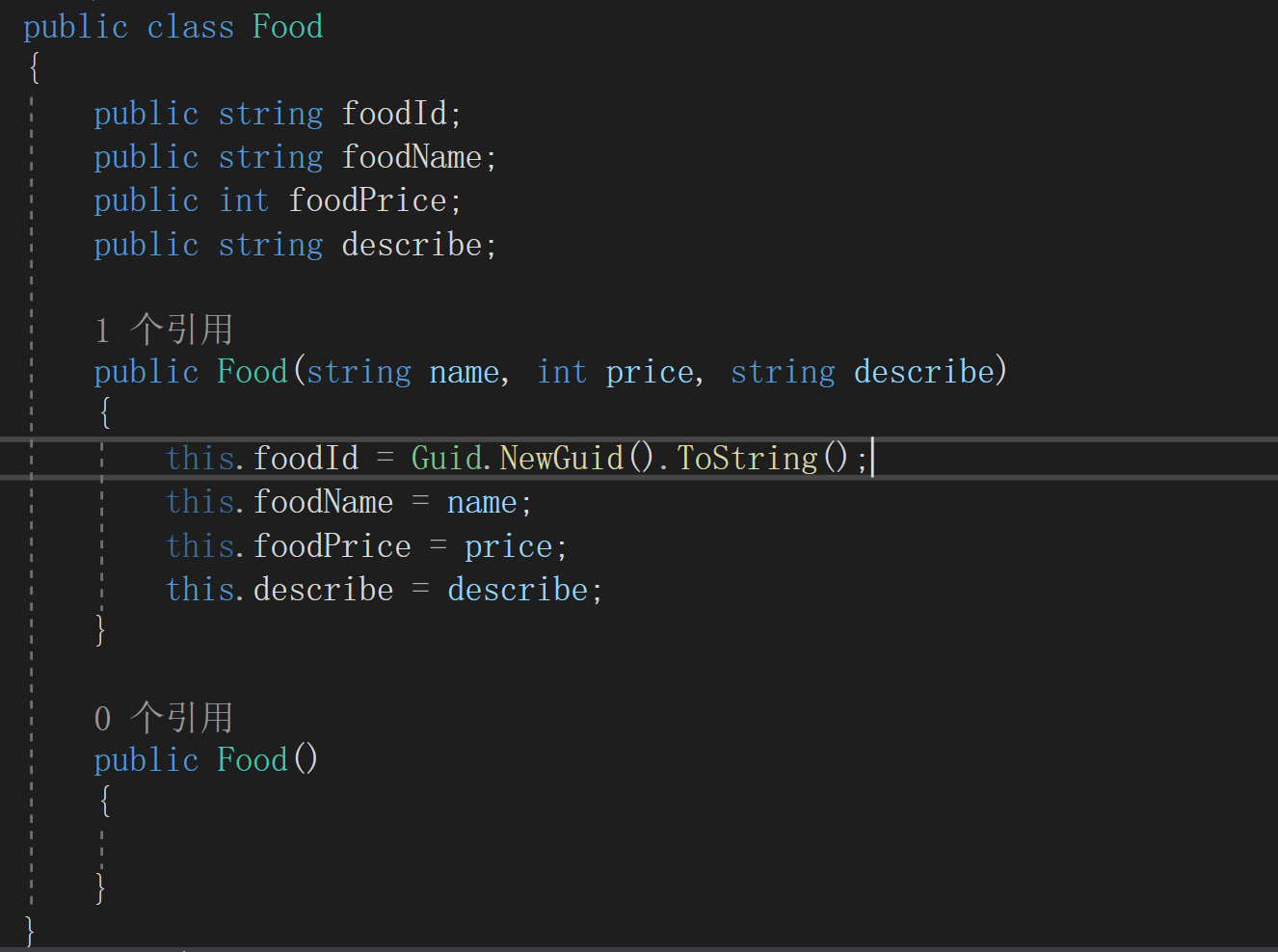
观察者模式的优点：

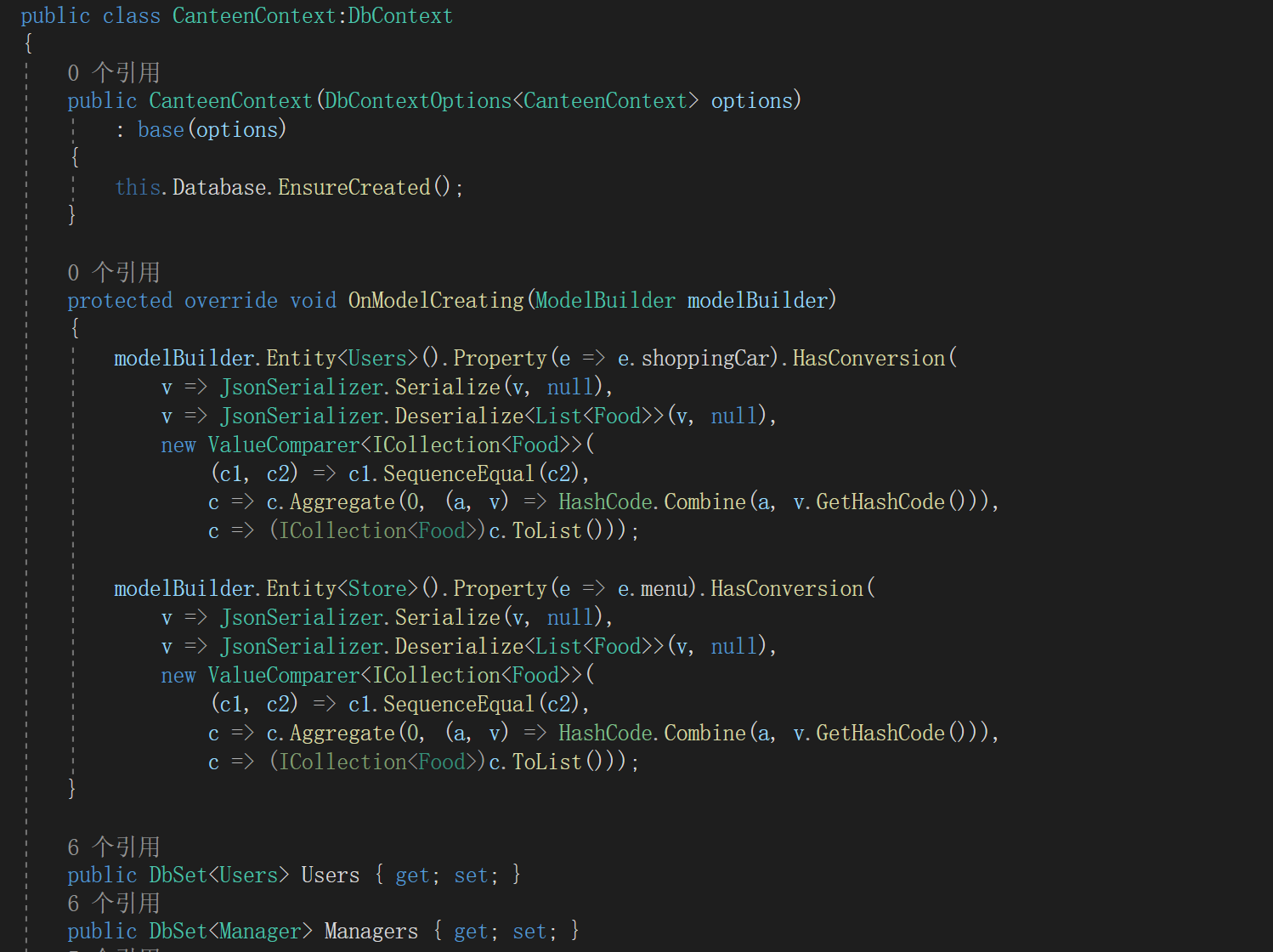
* 可以实现表示层和数据逻辑层的分离；
* 在观察目标和观察者之间建立一个抽象的耦合；
* 支持广播通信，简化了一对多系统设计的难度；
* 符合开闭原则，增加新的具体观察者无需修改原有系统代码，在具体观察者与观察目标之间不存在关联关系的情况下，增加新的观察目标也很方便；

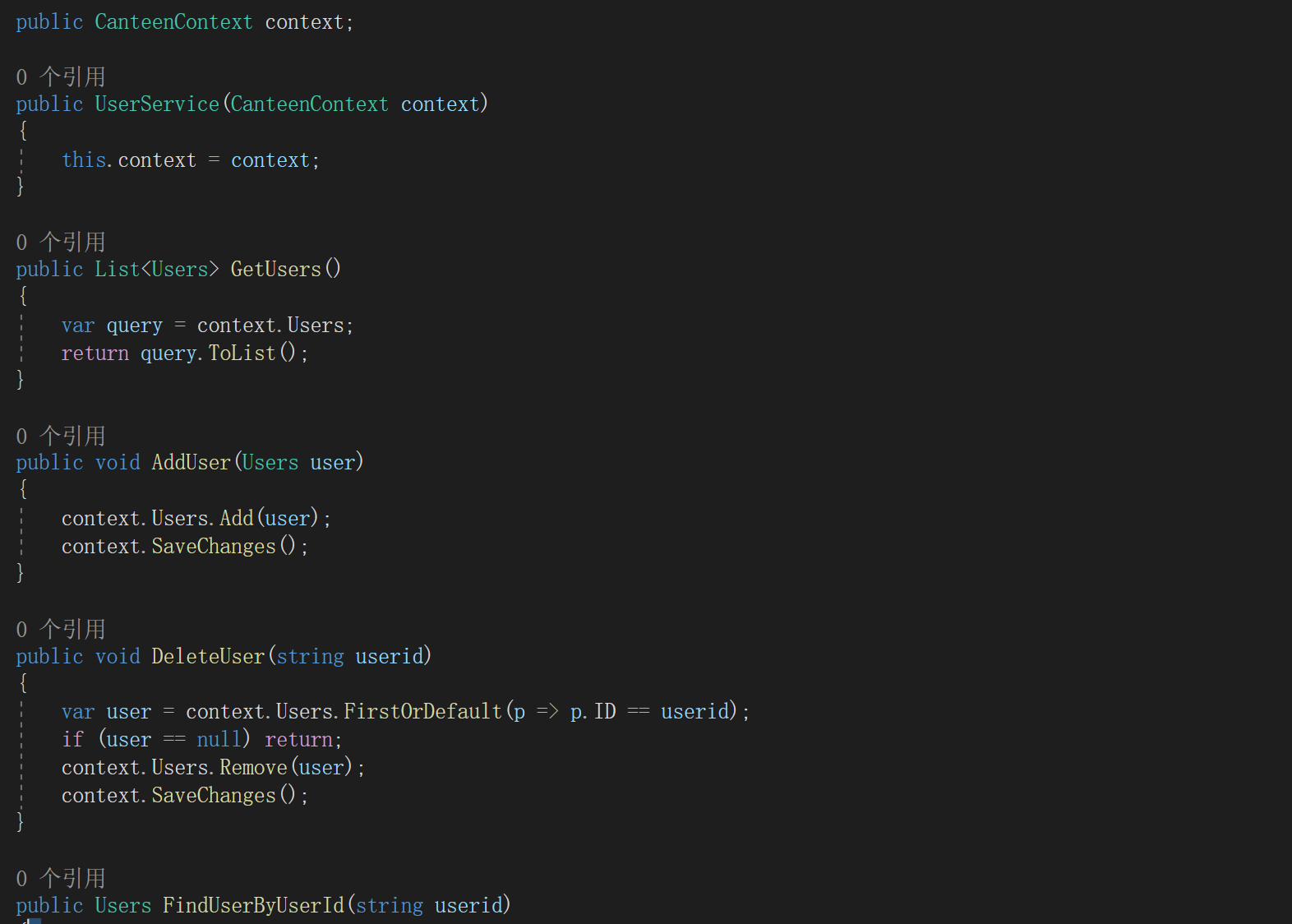


我们在餐饮系统中也运用了观察者模式，其中的MVC架构是我们在系统中主要采用的，通过模型，视图和控制器可以很方便的进行前后端分离，视图主要是前端的部分（但我们不太会写所以暂时没有写），而模型主要存储的是我们的类，我们还建了一个服务的文件夹，这里面写的是每个类对于数据库的函数，最后控制器主要是写Web Api的，可以通过控制器来进行前后端数据的通信。并且当模型层的数据发生改变的时候，视图层将自动改变其显示的内容。









**二、实现功能**

我们在各个类的Service中实现了对于数据库的增删改查，主要想要实现，顾客把食物放入购物车之后，购买的功能，模拟我们在淘宝上或者外卖平台上的选购和购买的功能。这个功能的本质就是对于数据库的增删改查。

