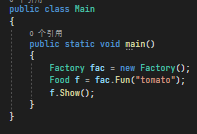
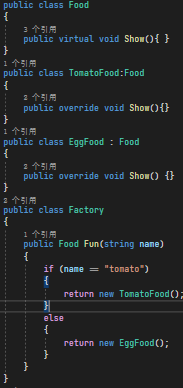
1. 工厂模式

所有的对象的创造都有一个对象去创造。



1. 策略模式

通过对算法进行封装，把使用算法的责任和算法的实现分割开来，并委派给不同的对象对这些算法进行管理。

1. 观察者模式

当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。这种模式有时又称作发布-订阅模式、模型-视图模式

1. 代理模式

委托，在某些情况下，一个对象不适合或者不能直接引用另一个对象，而代理对象可以在客户端和目标对象之间起到中介的作用。

1. 单例模式

一个类只有一个实例，且该类能自行创建这个实例的一种模式。

1. 门面模式

一个类管理大量不相干的类

1. 建造者/责任链模式

将一个复杂对象的构造与它的表示分离，使同样的构建过程可以创建不同的表示

1. 中介者模式

一个中介对象来封装一系列对象之间的交互，使原有对象之间的耦合松散，且可以独立地改变它们之间的交互。

迪米特原则：如果两个软件实体无须直接通信，那么就不应当发生直接的相互调用，可以通过第三方转发该调用。

1. 状态模式

对有状态的对象，把复杂的“判断逻辑”提取到不同的状态对象中，允许状态对象在其内部状态发生改变时改变其行为。