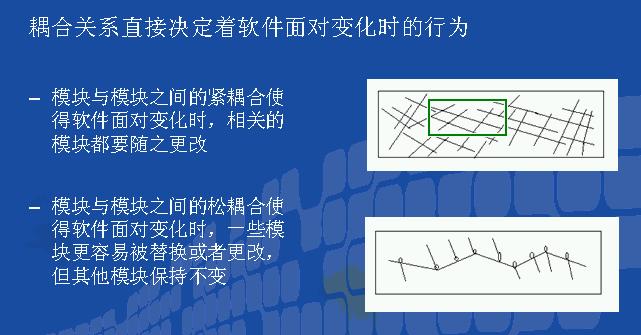
耦合关系



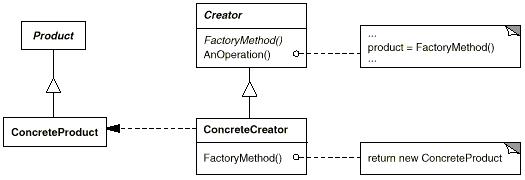
动机：

在软件系统中，由于需求的变化，“这个对象的具体实现”经常面临着剧烈的变化，但它却有比较稳定的接口。

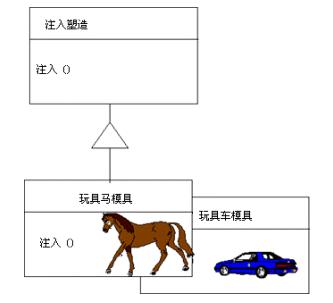
如何应对这种变化呢？提供一种封装机制来隔离“这个易变对象”的变化，从而保持系统中“其他依赖对象”不随需求的变化而变化。

意图：

定义一个用户创建对象的接口，让子类决定实例哪个类。Factory Method使一个类的实例化延迟到子类。

结构图：  
 

生活实例：

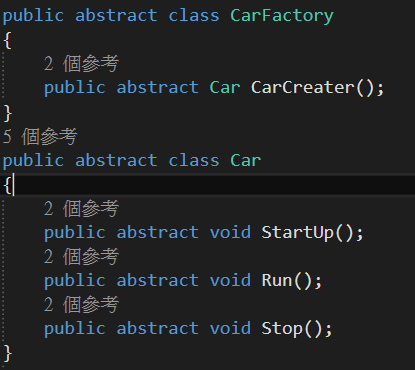


实例代码：

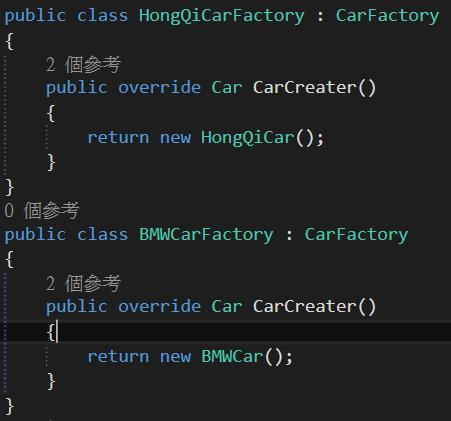
1. 当一个类不知道它所必须创建的对象类的时候
2. 当一个类希望由它子类来指定它所创建的对象的时候
3. 当类将创建对象的职责委托给多个帮助子类中的某个，并且你希望将哪个帮助子类是代理者这一信息局部化的时候

实例代码：

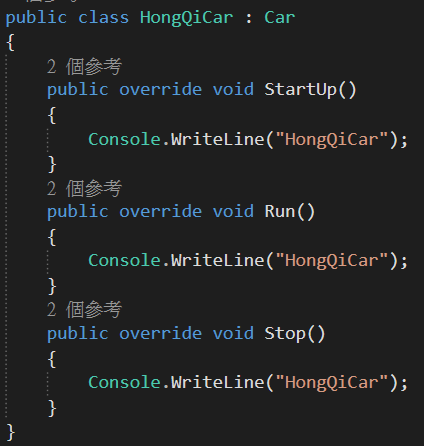
CarFactory类&Car类：

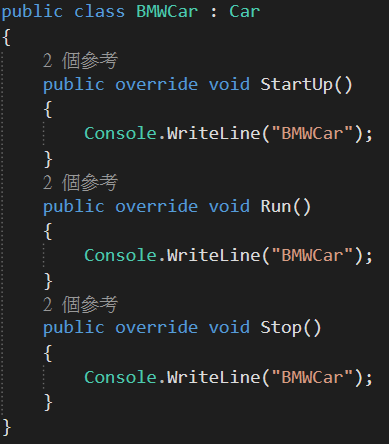


具体车型Factory类：

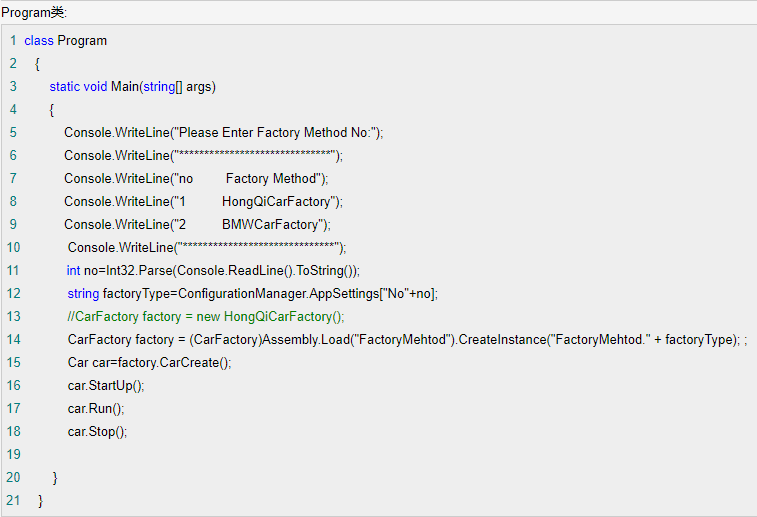


具体车型Car类：





Program类：



Factory Method模式的几个要点：

Factory Method模式主要用于隔离类的对象的使用者和具体类型之间的耦合关系。面对一个经常变化的具体类型，紧耦合关系会导致软件的脆弱。

Factory Method模式通过面向对象的手法，将所要创建的具体对象工作延迟到子类，从而实现一种扩展（而非改变）的策略，较好的解决了这种紧耦合关系。

Factory Method 模式解决“单个对象”的需求变化（当创建的实例不同的时候），Abstract Factory模式解决“系列对象”的需求变化（当需要创建新的类型的对象的时候），Builder模式解决“对象部分”的需求变化（当对象内部的方法需要变化的时候）。