工厂模式：

简单工厂模式

在工厂类里实例化相应的子类（把一个类的实例化放在另一个类的静态方法里），创建实例类的方法为静态方法。

工厂方法模式

定义一个用于创建对象的接口，让子类决定实例化哪一个类，工厂方法使一个类的实例化延其子类。

创建一个接口工厂，创建实现这个接口的工厂类（有多个、这里继承接口，并实例化相应需要实例化的类）

在C#的学习中，容易混淆virtual方法和abstract方法的使用，现在来讨论一下二者的区别。二者都牵涉到在派生类中与override的配合使用。

一、Virtual方法（虚方法）

     virtual 关键字用于在基类中修饰方法。virtual的使用会有两种情况：

     情况1：在基类中定义了virtual方法，但在派生类中没有重写该虚方法。那么在对派生类实例的调用中，该虚方法使用的是基类定义的方法。

     情况2：在基类中定义了virtual方法，然后在派生类中使用override重写该方法。那么在对派生类实例的调用中，该虚方法使用的是派生重写的方法。

　　情况3：多态情况，在基类中定义virtual，然后在派生类中使用override重写该方法。那么在多态（基类 变量=new 派生类（））的时候，调用的还是派生类重写的方法。

二、Abstract方法（抽象方法）

     abstract关键字只能用在抽象类中修饰方法，并且没有具体的实现。抽象方法的实现必须在派生类中使用override关键字来实现。

三、基类和派生类中都定义了一个相同的方法，这时候，实例化时，派生类前加new 关键字，可以隐藏基类方法，多态调用基类方法，直接实例化派生类，不会执行基类方法

联盟快卖 商人，