# 单例模式

# 工厂模式：

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迟到其子类；

优点：

首先，良好的封装性，代码结构清晰。一个对象创建是有条件约束的，如一个调用者需

要一个具体的产品对象，只要知道这个产品的类名（或约束字符串）就可以了，不用知道创

建对象的艰辛过程，降低模块间的耦合。

其次，工厂方法模式的扩展性非常优秀。在增加产品类的情况下，只要适当地修改具体

的工厂类或扩展一个工厂类，就可以完成“拥抱变化”。例如在我们的例子中，需要增加一个

棕色人种，则只需要增加一个BrownHuman类，工厂类不用任何修改就可完成系统扩展。

再次，屏蔽产品类。这一特点非常重要，产品类的实现如何变化，调用者都不需要关

心，它只需要关心产品的接口，只要接口保持不变，系统中的上层模块就不要发生变化。因

为产品类的实例化工作是由工厂类负责的，一个产品对象具体由哪一个产品生成是由工厂类

决定的。在数据库开发中，大家应该能够深刻体会到工厂方法模式的好处：如果使用JDBC

连接数据库，数据库从MySQL切换到Oracle，需要改动的地方就是切换一下驱动名称（前提

条件是SQL语句是标准语句），其他的都不需要修改，这是工厂方法模式灵活性的一个直接

案例。

最后，工厂方法模式是典型的解耦框架。高层模块值需要知道产品的抽象类，其他的实

现类都不用关心，符合迪米特法则，我不需要的就不要去交流；也符合依赖倒置原则，只依

赖产品类的抽象；当然也符合里氏替换原则，使用产品子类替换产品父类，没问题！、

# 抽象工厂模式