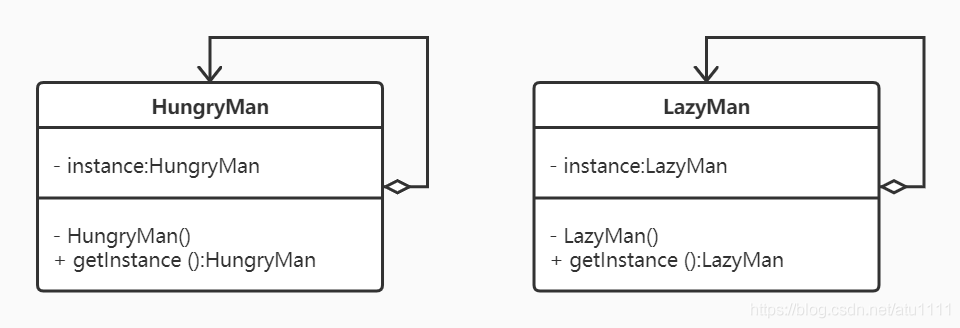
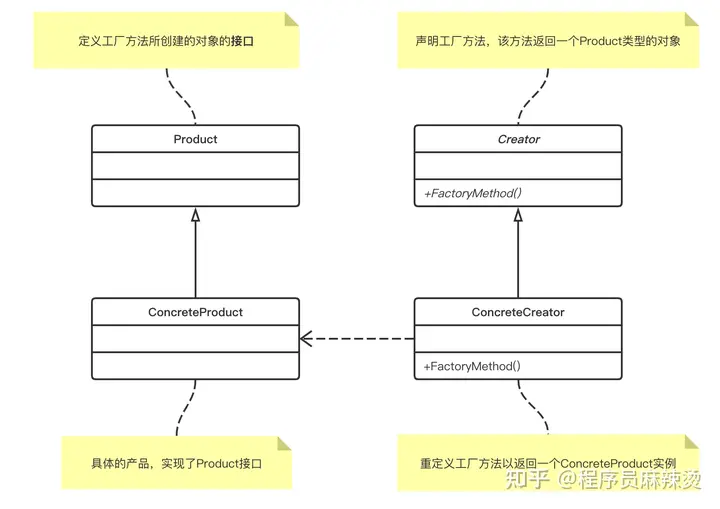
**单例模式：**



指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。例如，Windows 中只能打开一个任务管理器，这样可以避免因打开多个任务管理器窗口而造成内存资源的浪费，或出现各个窗口显示内容的不一致等错误。

  在计算机系统中，还有 Windows 的回收站、操作系统中的文件系统、多线程中的线程池、显卡的驱动程序对象、打印机的后台处理服务、应用程序的日志对象、数据库的连接池、网站的计数器、Web 应用的配置对象、应用程序中的对话框、系统中的缓存等常常被设计成单例。

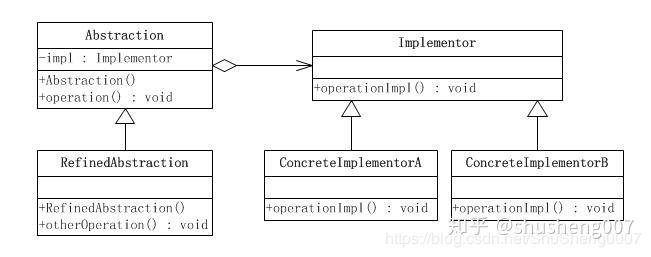
**工厂模式：**



工厂模式一般用于对于不同的场景，需要创建不同的对象，但是这些对象实现的功能是很相似的，可以抽象出一个父类，对于这种情形就可以使用工厂模式。

在实际中，很多框架都支持多种配置文件，项目启动时解析配置文件，将文件内容写入到内存中。配置文件格式很多，有xml、json、yaml等，这个时候就需要根据后缀来解析文件，使用工厂模式就很合理。

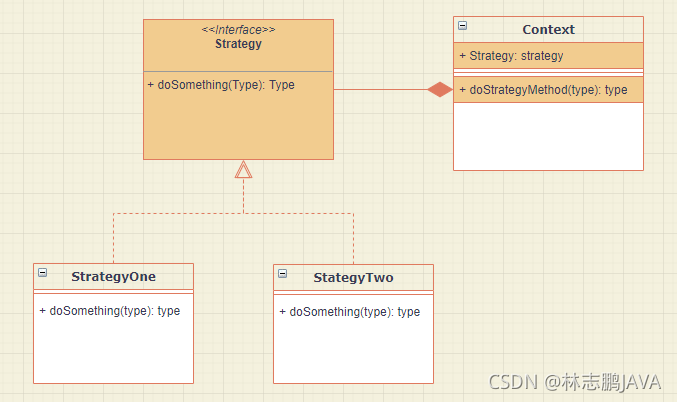
**桥接模式：**



以一个现实情况为例，画圆和画三角形，圆和三角形有绿色、红色以及黄色，如果使用继承的方式，那就会出现6个子类。如果使用组合的方式，只需要2+3=5个类，减少了类的创建。

桥接模式的抽象与实现两者之间是完全独立，假设再要加新的颜色或者图形，直接增加代码即可。

**策略模式：**



策略模式简单理解，应该是对于同一个业务功能，在不同的场景需求下提供不同的实现逻辑，来达到动态切换业务算法，满足不同场景的目的。同时它也有另外的好处，即优化代码结构，使其脱离大量逻辑判断，对外只提供上下文，让算法与实际业务代码解耦，对使用者屏蔽底层实现逻辑。