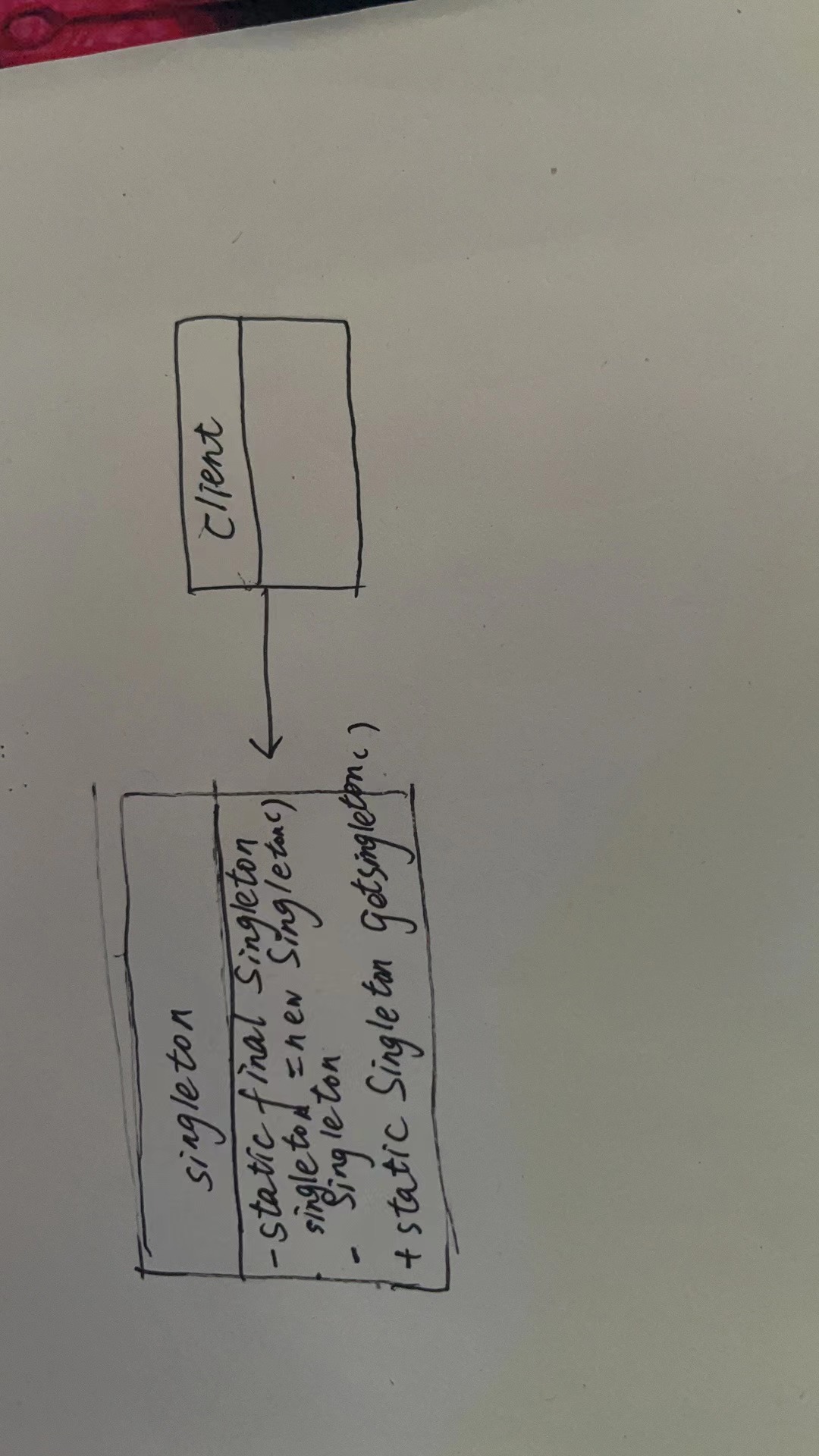
**单例模式：**

单例模式是一种创建型模式，它保证一个类仅有一个实例，并提供一个访问它的全局访问点。使用场合包括：

当类只能有一个实例而且客户可以从一个众所周知的访问点访问它时。

当这个唯一实例应该是通过子类化可扩展的，并且客户应该无需更改代码就能使用一个扩展的实例时



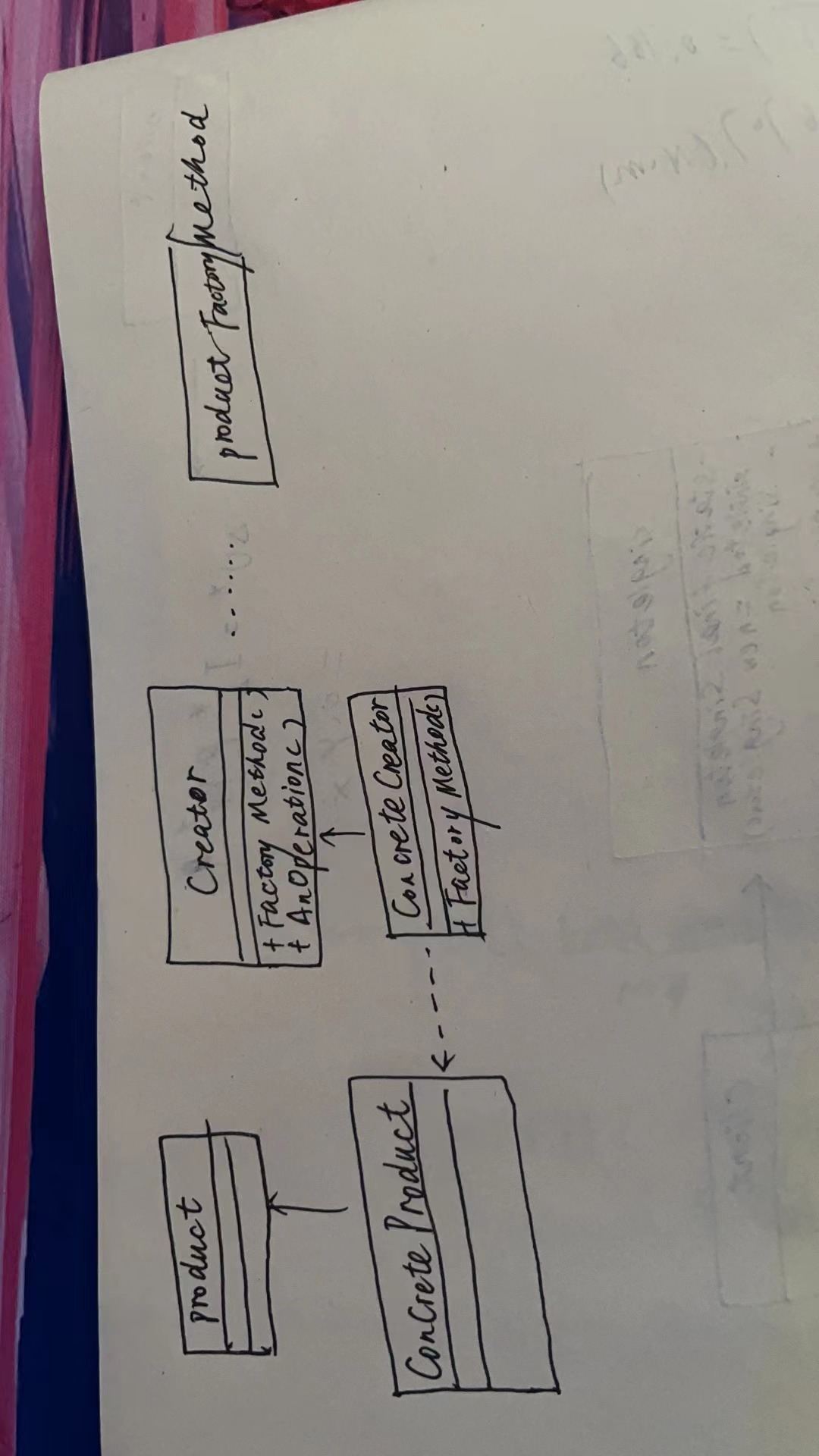
**工厂方法模式：**

工厂方法模式是一种创建型模式，它定义了一个用于创建对象的接口，让子类决定将哪一个类实例化。工厂方法模式使一个类的实例化延迟到其子类。使用场合包括：

当一个类不知道它所必须创建的对象的类的时候。

当一个类希望由它的子类来指定它所创建的对象的时候。

当类将创建对象的职责委托给多个帮助子类中的某一个，并且希望将哪一个帮助子类是代理者这一信息局部化的时候。

****

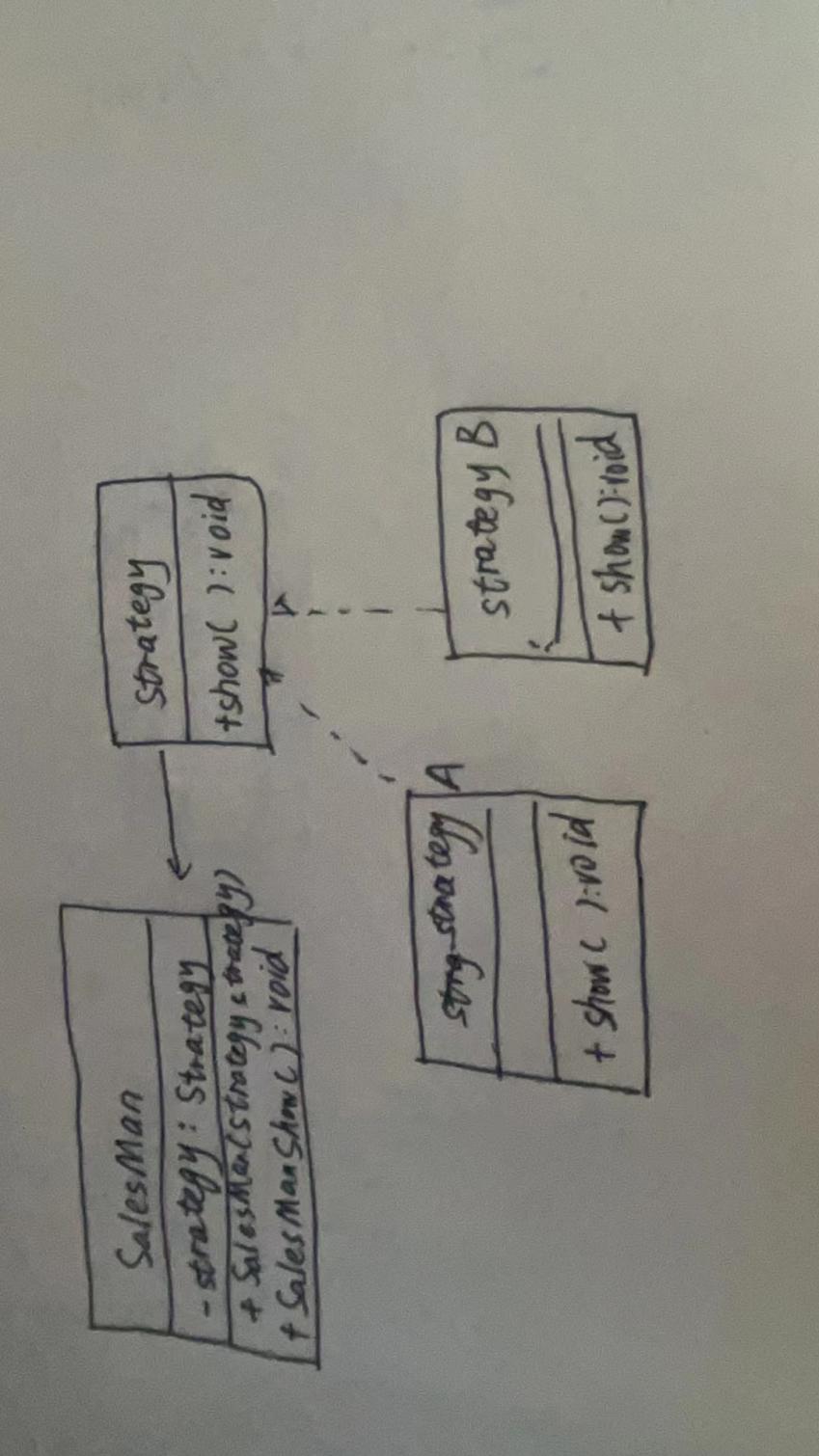
**策略模式：**

策略模式是一种行为型模式，它定义了算法家族，分别封装起来，让它们之间可以互相替换，此模式让算法的变化独立于使用算法的客户。使用场合包括：

当你有许多类似的类，它们只是在行为上稍有不同的时候。

当你需要在运行时选择算法的不同变体时。

当你有许多不同的算法，并希望能够轻松地切换算法时。



**桥接模式：**

桥接模式： 桥接模式是一种结构型模式，它将抽象部分与它的实现部分分离，使它们都可以独立地变化。使用场合包括：

当你不希望在抽象和它的实现部分之间有一个固定的绑定关系时。

当你希望在抽象部分和它的实现部分之间增加多层次的继承关系时。

当你希望在实现部分中的一个类可以有多种变化时。

