**策略模式(strategy)**

策略模式是一种对象行为型模式。一般对应于解决某一个问题的一个算法族，允许用户从该算法族中任选一个算法解决某一个问题，同时可以方便的更换算法或者增加新的算法。并且由客户端决定调用哪个算法。

**本质: 分离算法，选择实现**

*--以下摘自 http://www.runoob.com*

**意图：**定义一系列的算法,把它们一个个封装起来, 并且使它们可相互替换。

**主要解决：**在有多种算法相似的情况下，使用 if...else 所带来的复杂和难以维护。

**何时使用：**一个系统有许多许多类，而区分它们的只是他们直接的行为。

**如何解决：**将这些算法封装成一个一个的类，任意地替换。

**关键代码：**实现同一个接口。

**应用实例：** 1、诸葛亮的锦囊妙计，每一个锦囊就是一个策略。 2、旅行的出游方式，选择骑自行车、坐汽车，每一种旅行方式都是一个策略。 3、JAVA AWT 中的 LayoutManager。

**优点：** 1、算法可以自由切换。 2、避免使用多重条件判断。 3、扩展性良好。

**缺点：** 1、策略类会增多。 2、所有策略类都需要对外暴露。

**使用场景：** 1、如果在一个系统里面有许多类，它们之间的区别仅在于它们的行为，那么使用策略模式可以动态地让一个对象在许多行为中选择一种行为。 2、一个系统需要动态地在几种算法中选择一种。 3、如果一个对象有很多的行为，如果不用恰当的模式，这些行为就只好使用多重的条件选择语句来实现。

**注意事项：**如果一个系统的策略多于四个，就需要考虑使用混合模式，解决策略类膨胀的问题。

*--摘要结束*

关于策略模式(strategy)与工厂模式(factory)，两者确实比较相似，以至于我刚接触时感到很疑惑，在经过一番资料的查找后终于渐渐明了…其实还真差不多。你用其中一个能完成的功能用另一个也能完成的有模有样的。**两者合二为一，那更是牛B，可以在Context上下文对象中使用factory来实例化strategy对象。这样调用者只需要和一个类进行交互即可实现选择不同策略的功能。**

分类举例的话:

* 工厂模式是对象创建型模式，专门用于创建对象的，他根据需求创建出不同的实例，将对象的创建和调用者进行了解耦，解决了资源的统一分发。
* 策略模式是对象行为型模式，作用是让一个对象在许多行为中选择一种行为执行，他将策略的变化和使用策略的对象进行了解耦

简单来说工厂模式封装对象，实例化对象后调用的时候要知道具体的方法，策略模式封闭的是行为，调用的时候必须先制定实例化具体的类，再调用抽象的方法。