

策略模式

概念

在策略模式（Strategy Pattern）中，一个类的行为或其算法可以在运行时更改。这种类型的设计模式属于行为型模式。

在策略模式中，我们创建表示各种策略的对象和一个行为随着策略对象改变而改变的 context 对象。策略对象改变 context 对象的执行算法。

实施方案

比如定义一系列的算法,把每一个算法封装起来,并且使它们可相互替换，使得算法可独立于使用它的客户而变化

把一个类中经常改变或者将来可能改变的部分提取出来，作为一个接口，然后在类中包含这个对象的实例，这样类的实例在运行时就可以随意调用实现了这个接口的类的行为

适用情况

“策略”提供了一种用多个行为中的一个行为来配置一个类的方法。即一个系统需要动态地在几种算法中选择一种

当一个应用程序需要实现一种特定的服务或者功能，而且该程序有多种实现方式时使用

一个类定义了多种行为，并且这些行为在这个类的操作中以多个条件语句的形式出现

优点

可以动态的改变对象的行为

避免使用多重条件判断

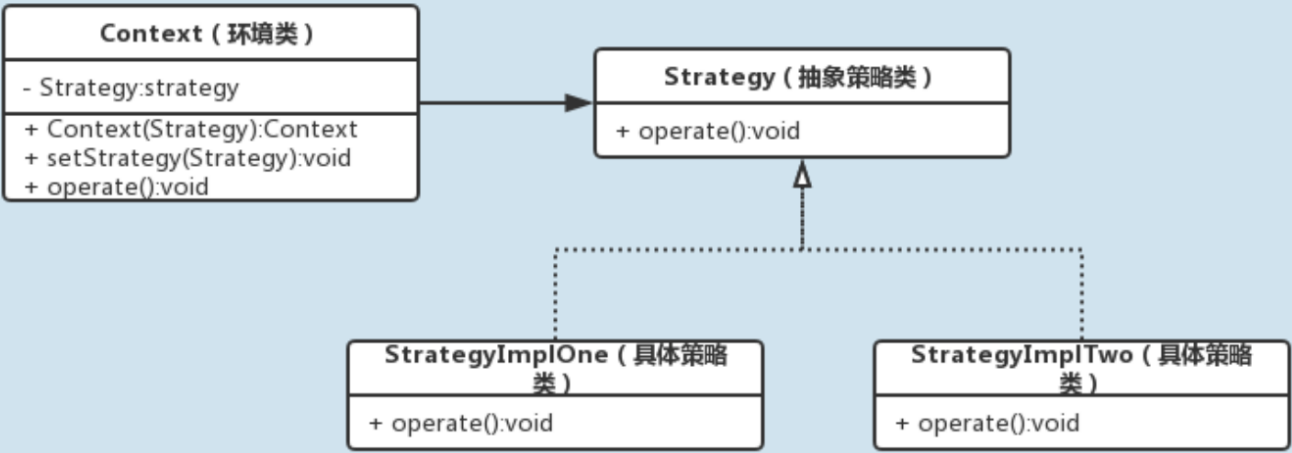
扩展性良好

缺点

客户端必须知道所有的策略类，并自行决定使用哪一个策略类

策略模式将造成产生很多策略类

所有策略类都需要对外暴露



UML