

Java SE程序设计 北京圣思园科技有限公司

主讲人 张龙

All Rights Reserved



策略模式(Strategy)

- 策略模式（**Strategy Pattern**）中体现了两个非常基本的面向对象设计的原则
 - 封装变化的概念
 - 编程中使用接口，而不是对接口的实现
- 面向接口的编程



策略模式(Strategy)

- 策略模式的定义
 - 定义一组算法，将每个算法都封装起来，并且使它们之间可以互换。
 - 策略模式使这些算法在客户端调用它们的时候能够互不影响地变化



策略模式(Strategy)

- 策略模式的意义

- 策略模式使开发人员能够开发出由许多可替换的部分组成的软件，并且各个部分之间是**弱连接**的关系。
- 弱连接的特性使软件具有更强的可扩展性，易于维护；更重要的是，**它大大提高了软件的可重用性**



策略模式(Strategy)

- 策略模式的组成
 - **抽象策略角色**：策略类，通常由一个**接口**或者**抽象类**实现
 - **具体策略角色**：包装了相关的算法和行为
 - **环境角色**：持有一个**策略类的引用**，最终给客户端调用的。



策略模式(Strategy)

- 策略模式的实现

- 策略模式的用意是针对一组算法，将每一个算法封装到具有共同接口的独立的类中，**从而使它们可以相互替换。**
- 策略模式使得算法可以在不影响到客户端的情况下发生变化。**使用策略模式可以把行为和环境分割开来。**
- 环境类负责维持和查询行为类，各种算法则在具体策略中提供。**由于算法和环境独立开来，算法的修改都不会影响环境和客户端**



策略模式(Strategy)

- 策略模式的编写步骤
 - 1. 对策略对象定义一个公共接口。
 - 2. 编写策略类，该类实现了上面的公共接口
 - 3. 在使用策略对象的类中保存一个对策略对象的引用。
 - 4. 在使用策略对象的类中，实现对策略对象的set和get方法（注入）或者使用构造方法完成赋值



策略模式(Strategy)

- 参看JDK Collections类的源代码



策略模式(Strategy)

- 实现自己的策略模式



策略模式(Strategy)

- 策略模式的缺点
 - 1.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。
 - 2.造成很多的策略类。



策略模式(Strategy)

- 解决方案
 - 采用工厂方法

