## 设计模式详述

## Observer

Observer Pattern定义了一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。主要解决一个对象状态改变给其他对象通知的问题，而且要考虑到易用和低耦合，保证高度的协作的问题。观察者与被观察者之间是属于轻度的关联关系，并且是抽象耦合的。

优点：1、观察者和被观察者是抽象耦合的。2、建立一套触发机制。

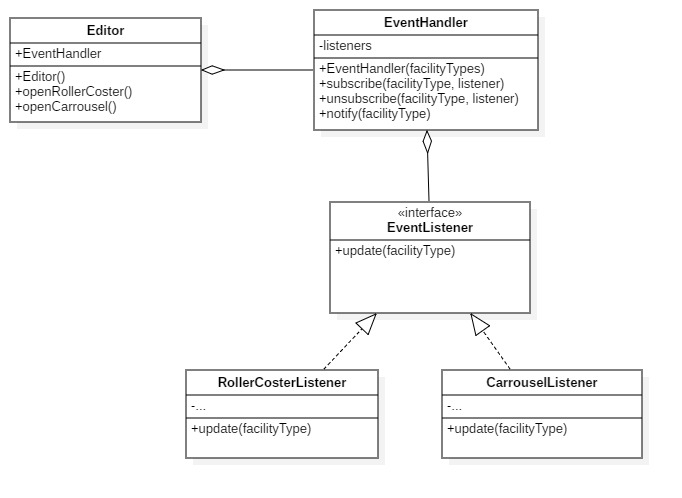
缺点：1、如果一个被观察者对象有很多的直接和间接的观察者的话，将所有的观察者都通知到会花费很多时间。2、如果观察者和观察目标之间有循环依赖的话，观察目标会触发它们之间进行循环调用，可能导致系统崩溃。 3、观察者模式没有相应的机制让观察者知道所观察的目标对象是怎么发生变化的，而仅仅只是知道观察目标发生了变化。

API描述

Observer场景为：迪士尼乐园有游乐设施如摩天轮和旋转木马，它们都还未开放，对这些设施感兴趣的游客们可以提前预约，当它们其中的项目开放或状态产生变化时，就会通知所有的预约者。

EventListener是用于处理Editor产生的状态改变，进而向特定的EventListener发送值得关注的事件，且允许新预约者加入和当前预约者离开列表的订阅框架。EventListener接口声明了通知接口，仅包含一个update()方法。RollerCosterListener和CarrouselListener分别对应了预约摩天轮和预约旋转木马的预约者。

Class diagram



## Filter

Filter Pattern（过滤器模式）又被称为Criteria Pattern（标准模式 ）允许开发人员使用不同的标准来过滤一组对象，通过逻辑运算以解耦的方式把它们连接起来。这种类型的设计模式属于结构型模式，它结合多个标准来获得单一标准。

优点：简单，解耦，使用方便。

缺点：在构建过滤规则的时候会有些繁琐。

API描述

Filter场景为：迪士尼乐园有游乐设施如摩天轮和旋转木马，每个游乐设施都对游客的年龄或身高有一定限制，在已预约该项目的游客中，需要以不同的条件筛选出符合游玩条件的游客并显示。

Criteria接口声明了标准接口，仅包含一个meetCriteria()方法。CriteriaAdult、CriteriaHarderHeight、CriteriaNormalHeight都定义了具体的简单标准，分别筛选年龄和不同标准的身高。RollerCosterCriteria和CarrouselCriteria分别定义了摩天轮和旋转木马游玩条件的复杂标准，用于筛选能够游玩该游乐设施的游客。

Class diagram

