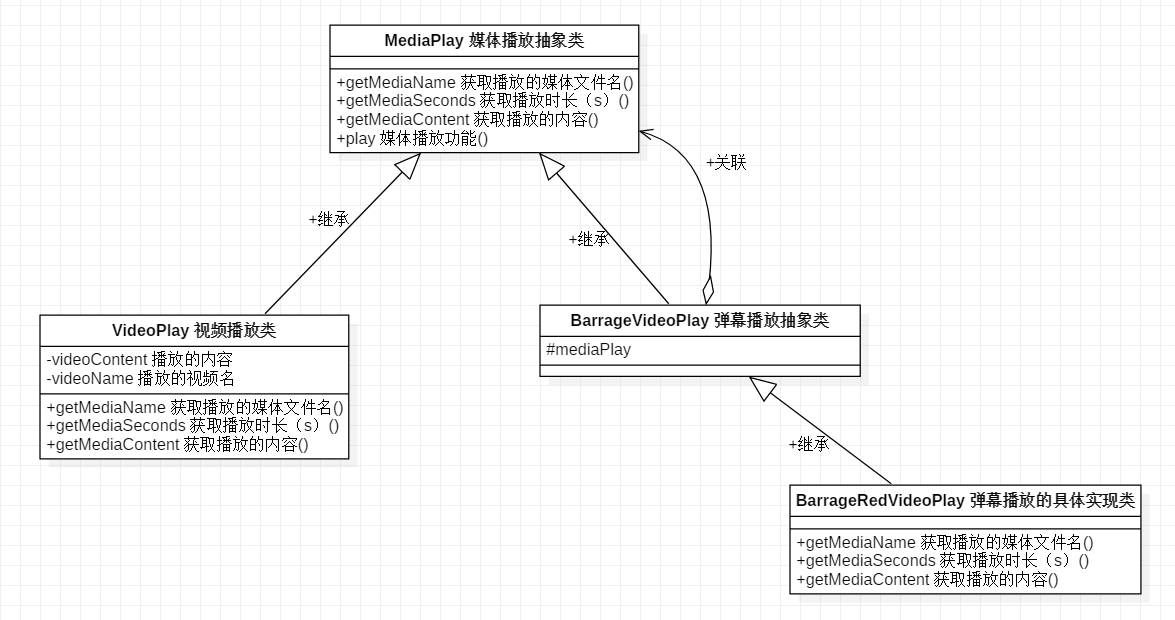
1. 定义

装饰器模式就是将原有的对象添加新的功能

1. 使用示例

我们给视频播放添加弹幕功能。



2.1 MediaPlay媒体播放抽象类



2.2 VideoPlay具体被装饰的类



2.3 BarrageVideoPlay弹幕播放抽象类

原有视频播放太过单调，我们在其基础上加入弹幕播放功能，BarrageVideoPlay抽象类用于定义我们要实现弹幕播放的功能，该类往往只是一个抽象类，内部持有MediaPlay类的引用，所以其子类可以比较方便的保留使用MediaPlay的已有功能。

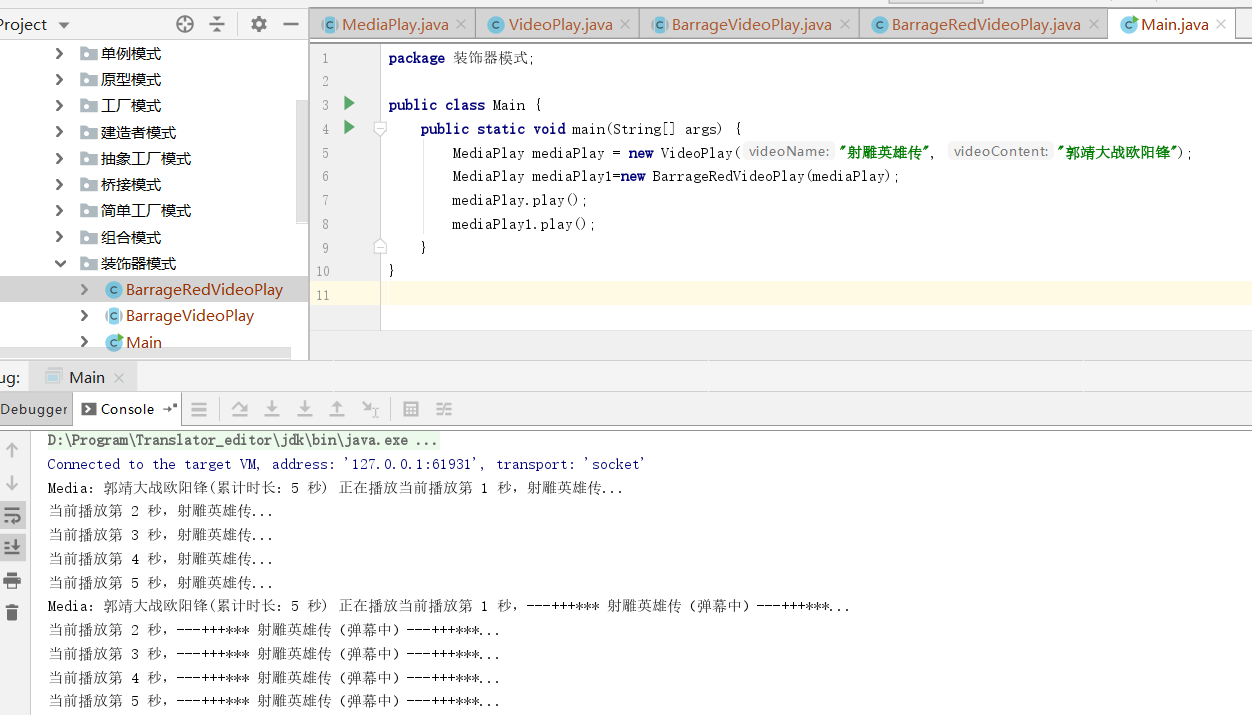


2.4 BarrageRedVideoPlay 弹幕播放具体实现类

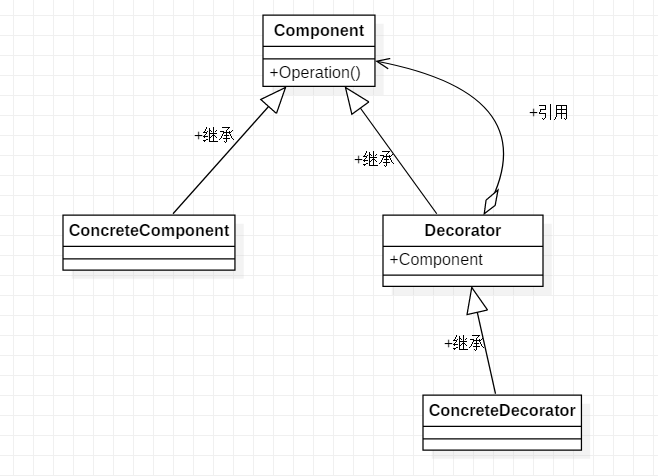
该类是将原有的媒体播放实现类传入，并将播放内容方法进行修饰。



测试类:



1. 组成角色



\*抽象构件(Component):

Component是一个接口或者抽象类，也是原始的对象，属于模式核心角色。用于定义一些抽象的接口或功能，以便后面的ConcreteComponment和ConcrereDecorator角色去实现。

\*具体构件(ConcreteComponment):

ConcreteComponment是最原始、最基本的接口或抽象类Component的实现，在模式中充当被修饰的角色。

\*抽象修饰角色(Decorator):

Decorator一般是一个抽象类、实现接口或者抽象方法，内部不一定有抽象方法定义，有可能只是单纯继承下Component抽象构件;但其内部一般都有一个Component角色的引用，表示Decorator需要装饰的对象，一般该对象是private或者protected声明。

\*具体装饰器角色(ConcreteDecorator):

具体的装饰器类，继承Decorator抽象装饰器角色，实现了Component抽象角色中定义的接口。

1. 优缺点

优点:装饰类与被装饰类可以独立发展，而不相互耦合。支持动态扩展原有类的功能。

缺点:装饰层级容易过多，导致系统复杂性增加。