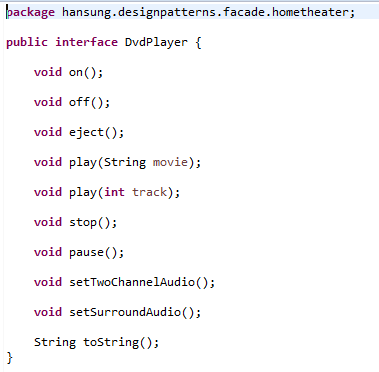
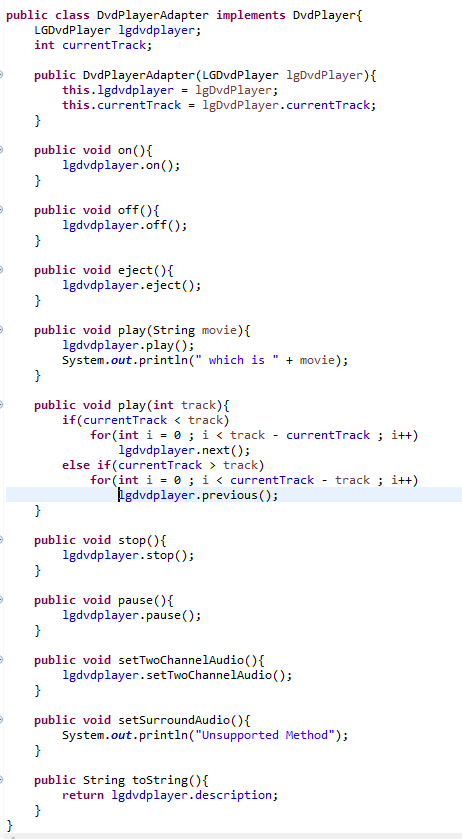
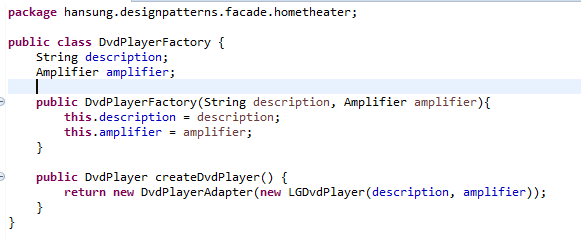
1. 객체 어댑터 패턴을 사용하여 DvdPlayer 클래스 대신에 LGDvdPlayer를 대체하시오.



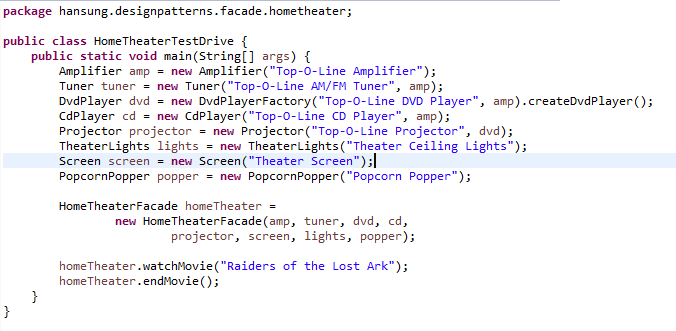
DvdPlayer Interface를 만들어주었다.



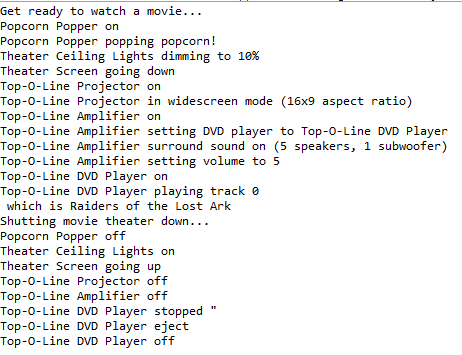
DvdPlayer interface를 구현한 DvdPlayerAdapter 클래스



팩토리 패턴을 이용하였습니다.

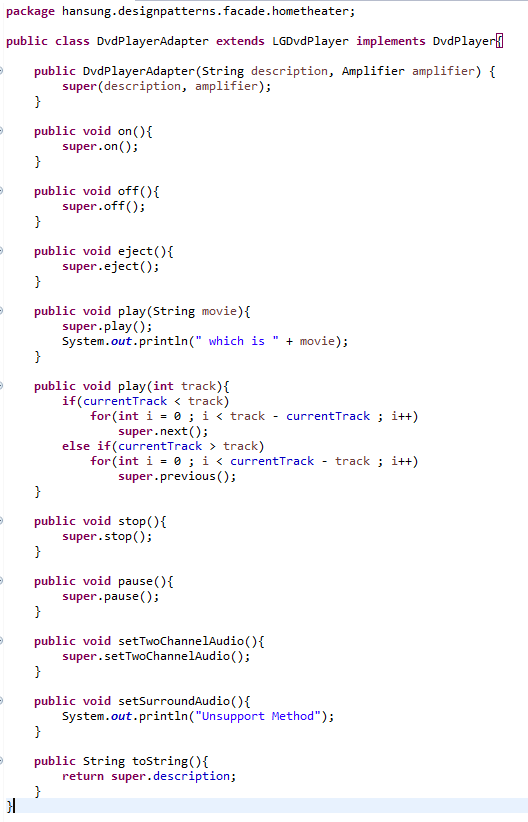


DvdPlayerFactory 클래스의createDvdPlater 메소드를 통해서 dvd객체 생성

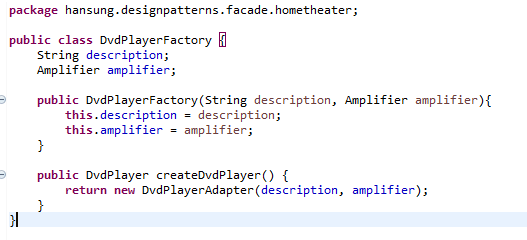


실행결과

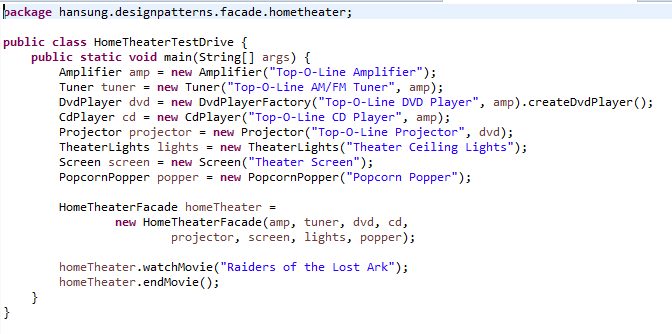
1. 클래스 어댑터 패턴을 사용하여 1번 문제를 해결하시오.

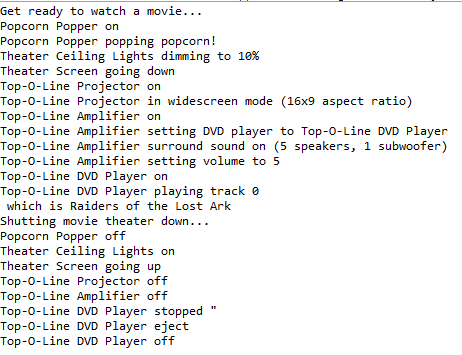


DvdPlayerAdapter 클래스를 LGDvdPlayer로부터 상속 받고 DvdPlayer 인터페이스를 구현



마찬가지로 팩토리 패턴을 이용하였으며 DvdPlayerAdapter의 구현 변화로 createDvdPlayer 메소드 수정





실행결과

1. 위의 실습결과를 바탕으로 객체 어댑터 패턴과 클래스 어댑터 패턴의 차이점에 대해서 토의 하시오.

객체 어댑터는 어댑티 클래스뿐만 아니라 그 서브 클래스에 대해서도 어댑터 역할을 수행할 수 있다. 서브클래스의 레퍼런스만 가지고 있으면 쉽게 이용할 수 있다

클래스 어댑터는 상속을 통하여 구현하며 특정 어댑티 클래스에서만 어댑터 역할을 수행하므로 어댑티의 서브클래스에 대해서는 어댑터 역할을 수행할 수 없다. 필요한 경우에 어댑티의 행위의 일부를 직접 오버라이드 할 수 있다. 어댑티로의 추가적인 객체 레퍼런스 없이 어댑티로 접근이 가능하다.