1. 싱글톤 패턴
   1. 개념 : 프로세스에서 어느 클래스의 특정 객체가 ‘하나만’ 존재해야 할 경우 사용.  
      (Ex-특정 어플리케이션의 다크모드, 어느 페이지를 들어가든 다크모드 동일적용)
   2. 작성법
      1. 싱글턴을 적용할 클래스 내에서 생성자를 private로 만들어, 다른 클래스에서 new 생성자로 호출하지 못하도록 막는다.
      2. static으로 자기자신 객체(변수)와 메소드를 만든다.  
         => 메모리의 지정된 공간에, 객체와 메소드 단 하나만 정적으로 만들기 위함
2. 전략 패턴
   1. 개념 : 프로세스의 각 기능별로 모듈을 분리하는것.  
      객체가 할 수 있는 행위 각각을 분리된 모듈(전략)로 만들면, 유지보수에 유연해진다.  
      => 행위의 수정이 필요할 경우 각각 모듈 내부만 수정하면 되기 때문!  
      (Ex-포털 검색화면에서, 이미지/뉴스/지도 등 선택모드에 따라 세분화된 검색 사용시)
   2. 작성법
      1. 공통 기능(ex-검색버튼 onclick)을 인터페이스로 구현한다.
      2. 모듈별로 해당 인터페이스를 implements한다.  
         => 각 모듈(클래스)들이 onclick과 연결!
3. State 패턴
   1. 개념 : 전략 패턴과 비슷하나, 전략 패턴이 기능별로 모듈을 분리하는 것과 달리 특정 상태마다 다르게 할 일을 모듈화하는 것.  
      유툽 다시보고 정리하자….
4. Command 패턴
   1. 개념 : 전략 패턴과 비슷하나, 전략 패턴이 기능별로 모듈을 분리하는 것과 달리 하는 일 자체를 다르게 하는 것.