1. 디자인패턴 - Builder패턴

* 생성자에 들어갈 매개변수에 관계 없이 차례대로 매개변수를 받고, 모든 매개변수를 받은 뒤 이것들로 인스턴스를 만드는 방식
* <https://jdm.kr/blog/217>

1. Effective Java의 Builder 패턴

* 인스턴스 생성시 각 인자가 어떤 값인지 알기 쉽다.(프로퍼티 명으로 구분가능)
* setter메소드가 없으므로 수정 불가능한 객체 생성 가능
* 생성자로 만드므로 일관성이 깨지지 않는다.
* builder클래스의 build 메서드에서 유효성 검증 코드를 넣을 수 있다.
* <https://johngrib.github.io/wiki/builder-pattern/>

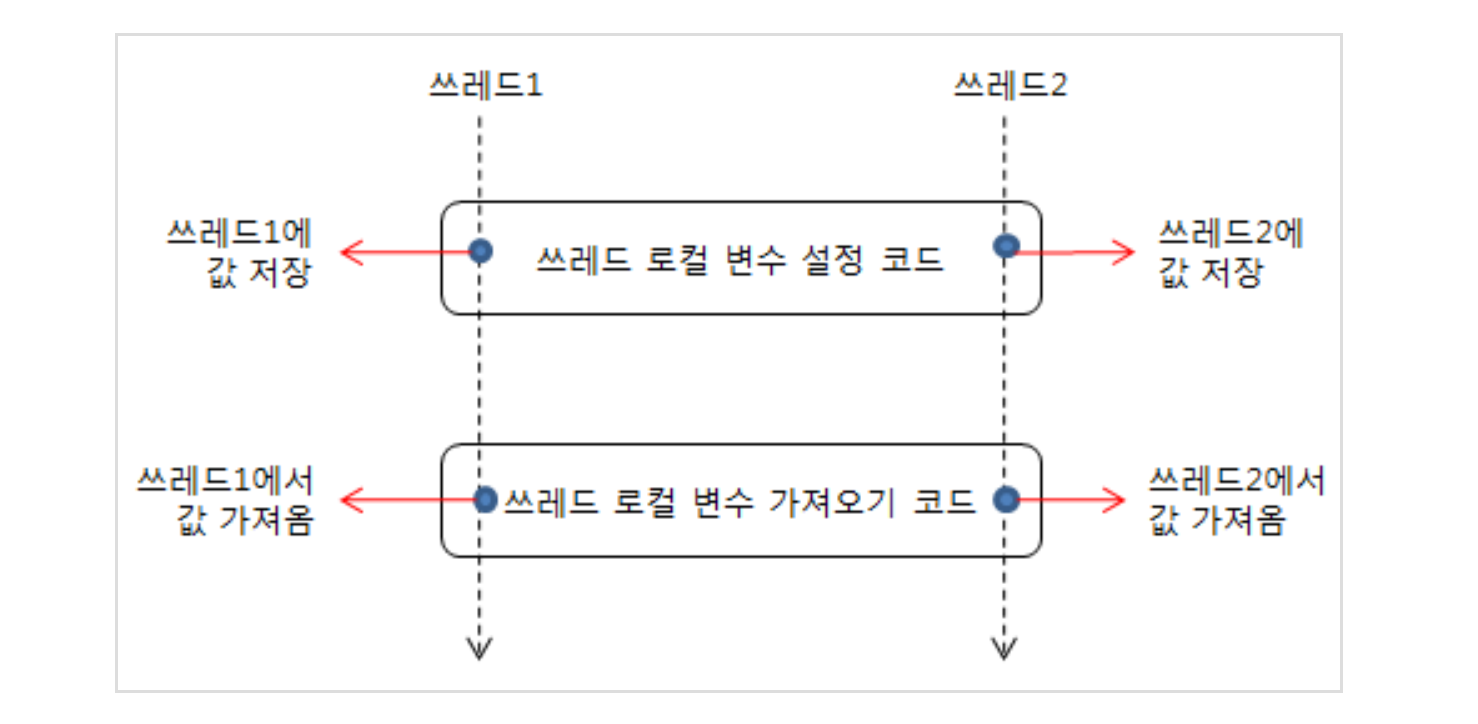
1. lombok으로 빌더패턴 쓰는법

* 애노테이션 @Builder를 추가하여 쉽게 사용 가능하다.
* 에노테이션을 붙여 이펙티브 자바 스타일과 비슷한 코드가 빌드된다.
* 생성자 파라메터 입력 및 build 메서드 수행으로 사용한다.

1. Java 웹 프로그래밍 4가지 scope(page, request, session application)
   1. Page : 하나의 JSP페이지(하나의 요청=하나의 페이지) 내에서만 객체를 공유하는 영역
   2. Request : 요청을 받아서 응답하기까지 객체가 유효한 영역
   3. Session : 같은 브라우저 내에서 같은 객체를 공유하는 영역, 세션종료시 반환됨
   4. Application : 같은 어플리케이션 내에서 객체를 공유하는 영역

* <http://victorydntmd.tistory.com/155>

1. 쓰레드 로컬

* 같은 스레드 안에서만 정보를 공유 할 수 있다.
* 
* 사용법
  + ThreadLocal 객체를 생성한다.
  + ThreadLocal.set() 메서드를 이용해서 현재 쓰레드의 로컬 변수에 값을 저장한다.
  + ThreadLocal.get() 메서드를 이용해서 현재 쓰레드의 로컬 변수 값을 읽어온다.
  + ThreadLocal.remove() 메서드를 이용해서 현재 쓰레드의 로컬 변수 값을 삭제한다.