* 'builder pattern' 이란?

점층적 생성자 패턴과 자바 빈 패턴의 장점을 결합한 것이 바로 빌더 패턴 이다

클라이언트 코드에서 필요한 객체를 직접 생성하는 대신, 그 전에 필수 인자들을 전달하어 빌더 객체를 만든 뒤, 빌더 객체에 정의된 설정 메서드들을 호출하여 인스턴스를 생성하는 것이다.

<http://mommoo.tistory.com/54>

<https://blog.seotory.com/post/2017/09/java-builder-pattern>

<http://asfirstalways.tistory.com/350>

* Java interface의 default 메소드 란?

Java 8에서는 디폴트 메서드라는 것을 사용하여 메서드 구현을 포함하는 인터페이스를 정의할 수 있다 인터페이스에서 이미 구현을 했으니 해당 인터페이스를 구현하는 클래스에서는 추가된 메서드의 구현을 추가적으로 할 필요가 없다 결과적으로 기존 인터페이스를 구현하는 클래스는 자동으로 인터페이스에 추가된 새로운 메서드의 디폴트 메서드를 상속받게 된다

<http://asfirstalways.tistory.com/353>

<http://blog.powerumc.kr/473>

<http://blog.eomdev.com/java/2016/03/30/default-method.html>

* java 웹 프로그래밍 - 4가지 scope ( page, request, session, application )

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

객체 범위 종류

웹 애플리케이션에는 4개의 객체 범위가 존재합니다.

( page 범위 / request 범위 / Session 범위 / Application 범위 )

1) page 영역

한 번의 브라우저 요청이 오면 하나의 JSP 페이지가 호출됩니다.

이 때 나의 JSP 페이지 내에서만 객체를 공유하는 영역을 의미합니다 ( JSP only )

- JSP 파일에는 pageContext가 내장되어 있으며, 이 객체는 page 영역에서만 유효합니다.

- JSP 파일에 <% %> 안에 변수를 사용하면 이 변수는 해당 JSP 파일 내에서만 유효합니다.

2) request 영역

요청을 받아서 응답하기까지 객체가 유효한 영역입니다.

- forward 또는 include를 사용하면 request 기본 객체가 공유되어서 request 영역이 됩니다.

- 지금까지 Servlet에서 JSP로 객체를 보낼 때 사용했던 방법입니다.

Servlet에서는 reqeust.setAttribute("이름", 객체); 로 전달하였고,

JSP에서는 Obejct ob j = request.getAttribute("이름"); 으로 받았습니다.

이 때 사용했던 request 객체가 request 영역입니다.

3) session 영역

하나의 브라우저 당 1개의 session 객체가 생성됩니다.

즉 같은 브라우저 내에서 요청되는 페이지들은 같은 객체를 공유하게 되는데 이를 세션 영역이라고 합니다.

- 세션이 종료되면 반환됩니다.

- requset.getSession() 메서드를 호출하여 세션 영역에서 유효한 객체를 얻을 수 있습니다.

4) application 영역

하나의 애플리케이션 당 1개의 application 객체가 생성됩니다.

즉 같은 애플리케이션 내에서 요청되는 페이지들은 같은 객체를 공유하게 되는데 이를 애플리케이션 영역이라고 합니다.

- 애플리케이션이 종료되면 반환

- request.getServletContext() 메서드를 호출하여 애플리케이션 영역에서 유효한 객체를 얻을 수 있습니다.

이로 미루어 보아 scope 범위는 page < request < session < application 순이라는 것을 알 수 있습니다.

