1. IoC컨테이너란? (Inversion of Control) : 제어 역전

일반적으로 프로그래밍을 작성할 때 프로그램이 흘러가는 흐름이나 생성되는 객체에 대한 제어권을 개발자가 만드는 개발자가 가지는 것과 달리 프레임워크가 가지는 것을 의미한다.

개발자가 코드의 흐름이나 객체 생성에 관련된 코드를 프로그래밍 코드에 직접 작성하는 것이 아닌 프레임워크가 사용하는 파일에 작성하면 이를 토대로 프레임워크가 객체를 생성하여 반환하고 코드가 동작하는 순서를 결정하게 된다는 의미이다.

IoC컨테이너의 종류: BeanFactory, ApplicationContext

* Spring 에서는 프로그램에서 사용할 객체를 bean configuration 파일에 정의하여 사용한다.
* class : 객체를 생성하기 위해 사용할 클래스를 지정한다.
* id : Bean 객체를 가져오기 위해 사용하는 이름을 지정한다.
* lazy-init : 싱글톤인 경우 xml을 로딩할 때 객체 생성 여부를 설정한다.   
   true : xml 로딩 시 객체를 생성하지 않고 객체를 가져올 때 생성한다.
* scope : 객체의 범위를 설정한다.  
   - singleton : 객체를 하나만 생성해서 사용한다.  
   - prototype : 객체를 가져올 때 마다 객체를 생성한다.

1. 빈에 값을 주입할 때 생성자 통한 주입은 ( )태그를 사용한다.

그 외 setter메소드를 통한 주입, 컬렉션, 자동주입인 ( )이 있다.

<bean id=*"t1" class="co.soft.beans.Test2">*

<constructor-arg value=*"100" type="int" index="2"/>*

<constructor-arg value=*"200" type="int" index="1"/>*

<constructor-arg value=*"300" type="int" index="0"/>*

</bean>

<bean id=*"data" class="co.soft.beans.Data scope="prototype"/>*

<bean id=*"t8" class="co.soft.beans.Test3">*

<constructor-arg ref=*"data"/>*

<constructor-arg ref=*"data"/>*

</bean>

1. AOP: Aspect Oriented Programming : 관점 지향 프로그래밍

* 하나의 프로그램을 관점(혹은 관심사)라는 논리적인 단위로 분리하여 관리하는 개념
* 로깅, 감사, 선언적 트랜젝션, 보안, 캐싱 등 다양한 곳에서 사용되고 있다.
* 관심사를 통해 Spring Framework가 어떤 메서드가 호출되는지 관심있게 지켜보다가 특정 메서드가 호출되면 자동으로 메서드 전과 후에 다른 메서드가 호출 될 수 있도록 한다.

1. @ModelAttribute의 역할은 무엇인가?

객체로 변수 주입가능

1. Model은 무엇인가?
2. ViewResolver의 역할은 무엇인가?



1. Redirect에 대해 아는대로 써라.  
   서버가 클라이언트에게 요청할 주소를 응답결과로 전달하는 것  
   클라이언트는 응답결과로 받은 요청주소를 직접 요청하게 되는데, 브라우저가 요청하는 것이므로 주소창의 주소는 변경이 된다.

Redirect는 새로운 요청이 발생하는 것으로 HttpServletRequest객체는 소멸 후 새롭게 생성되며 HttpSession객체는 그대로 유지된다.

1. Session에 대해 아는대로 써라.

브라우저가 최초로 서버에 요청을 하게 되면 브라우저당 하나씩 메모리 공간을 서버에서 할당하게 된다. 요청이 새롭게 발생하더라도 같은 메모리 공간을 사용하게 된다.

1. 프로그램 실행 중 변하지 않는 값이 있다면 properties 파일에 정의해서 사용할 수 있다.(O)
2. 유효성검사를 할 때 (@Valid)어노테이션을 쓴다. 유효성 검사 결과를 확인하고자할 때 (BindingResult)객체를 주입받아야 한다.