

## 1. 스프링의 빈

가. 어플리케이션을 형성하고, 스프링 IOC 컨테이너에 의해 관리되는 대상 객체

나. 컨테이너에 의해 인스턴스화되고 조합되는 객체

다. 이 빈들은 컨테이너에 제공되는 설정 메타파일로 생성

라. 예: XML의 `<bean/>`

## 2. 빈 정의 설정 메타정보 분류

가. 빈을 생성하는 법

나. 빈의 라이프사이클 상세

다. 빈의 의존성

## 3. 빈 정의 속성 상세

가. class: 필수요소로서 빈을 생성하는 빈 클래스를 지정

나. name: 유일한 빈 식별자를 지정, id/name 속성 사용

다. scope: 빈 정의로부터 생성된 오브젝트의 스코프를 지정

라. constructor-arg: 의존성을 주입하는데 사용

마. properties: 의존성을 주입하는데 사용

바. autowiring mode: 의존성을 주입하는데 사용

사. lazy-initialization mode: 요청받을 때 빈 인스턴스 생성토록 IOC 컨테이너에게 명령

아. initialization method: 빈에 필요한 프로퍼티에 설정된 후에 호출되는 콜백

자. destruction method: 빈을 포함한 컨테이너가 제거된 후에 사용되는 콜백

## 4. 메타정보를 설정하는 방법

가. XML 기반 설정 파일

나. 애노테이션 기반 설정

다. 자바기반 설정

## 5. XML 기반 설정 파일 예제

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.2.xsd">

<!-- A simple bean definition -->
<bean id="..." class="...">
<!-- collaborators and configuration for this bean go here -->
</bean>

<!-- A bean definition with lazy init set on -->
<bean id="..." class="..." lazy-init="true">
<!-- collaborators and configuration for this bean go here -->
</bean>

<!-- A bean definition with initialization method -->
<bean id="..." class="..." init-method="...">
<!-- collaborators and configuration for this bean go here -->
</bean>

<!-- A bean definition with destruction method -->
<bean id="..." class="..." destroy-method="...">
<!-- collaborators and configuration for this bean go here -->
</bean>

<!-- more bean definitions go here -->

</beans>
```