Bean lifecycle means, there are some methods is going to be call when the bean object will create and destroy.

There are 3 ways to manage bean lifecycle methods :

1. **Using XML**
2. **Using Annotation**
3. **By Implementing Interfaces**

The 3rd approach is least recommended i.e. **Implementing interface.** But we will still learn it for interview point of view.

Using Interfaces

There are two interfaces we need to implement in our bean class.

* **InitializingBean**
  + This interface contains a single method i.e. afterPropertiesSet() and this method is going to call by spring by default when object creation and all dependencies has been injected into the bean object.
* **DisposableBean**
  + This interface contains a single method i.e. destroy() and this method is going to call by spring by default when the bean object is going to be destroy.

**Person.java :**

public class Person implements DisposableBean, InitializingBean {

@Override

public void afterPropertiesSet() throws Exception {

System.out.println("Entry - Take entry coin");

}

public void chilling() {

System.out.println("Consuming alcohol, dancing and watching beauties");

}

@Override

public void destroy() {

System.out.println("Exit - Give back entry coin");

}

}

**Config.xml :**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context <http://www.springframework.org/schema/context/spring-context.xsd>">

<bean id="person" class="com.beanlifecycle.interfaces.Person" />

</beans>

**Bar.java :**

public class Bar {

public static void main(String[] args) {

ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("com/beanlifecycle/interfaces/config.xml");

Person p = context.getBean("person", Person.class);

p.chilling();

context.close();

}

}