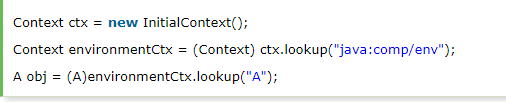
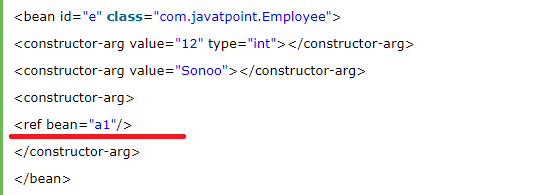
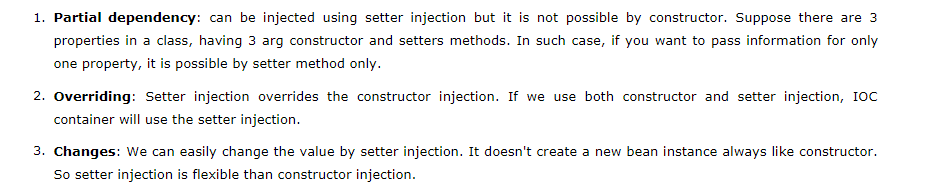
1. <p:propertyname=value>
2. <p:propertyname-ref=id>
3. It was **developed by Rod Johnson in 2003**.
4. It can be thought of as a *framework of frameworks* because it provides support to various frameworks such as Struts, Hibernate, Tapestry, EJB, JSF etc.
5. The framework, in broader sense, can be defined as a structure where we find solution of the various technical problems.
6. Context module supports internationalization (I18N), EJB, JMS, Basic Remoting.

Dependency Injection in Spring

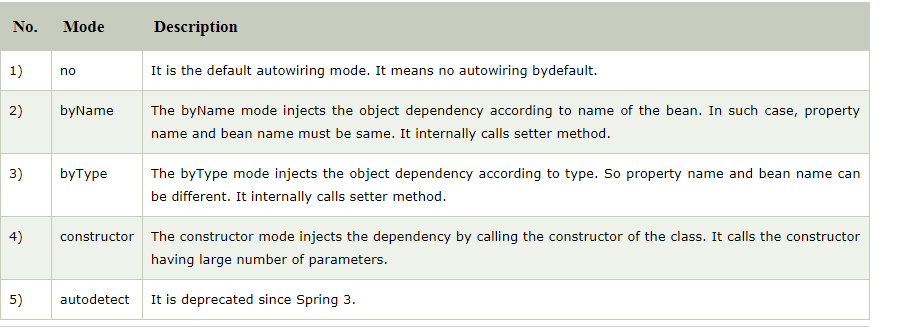
1. To understand the DI better, Let's understand the Dependency Lookup (DL) first: 🡺The Dependency Lookup is an approach where we get the resource after demand. There can be various ways to get the resource for example:
2. A obj = **new** AImpl();
3. A obj = A.getA();
4. Alternatively, we can get the resource by JNDI (Java Naming Directory Interface) as:**Problems of Dependency Lookup**
5. **tight coupling** The dependency lookup approach makes the code tightly coupled. If resource is changed, we need to perform a lot of modification in the code.
6. **Not easy for testing** This approach creates a lot of problems while testing the application especially in black box testing.



# Difference between constructor and setter injection

1. 

## **Advantage of Autowiring**

1. It requires the **less code**
2. The byName AUTOWIRING mode injects the object dependency according to name of the bean. In such case, property name and bean name must be same. It internally calls setter method.

#### **Why use AOP?**

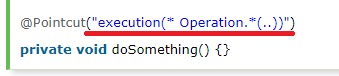
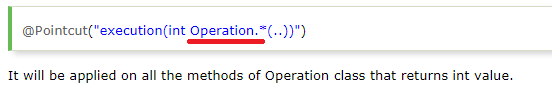
1. It provides the pluggable way to dynamically add the additional concern before, after or around the actual logic.

## **AOP Concepts and Terminology**

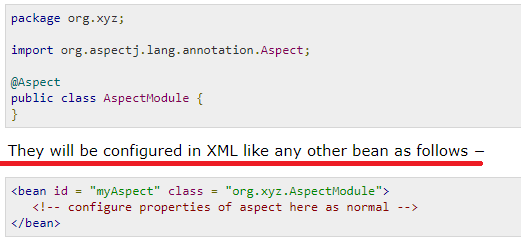
* Join point, Advice, Pointcut ,Introduction, Target Object, Aspect ,Weaving, Interceptor

Interceptor🡺it is an aspect that contains only one advice.

# Spring AOP AspectJ Annotation Example

1. there are two ways to use Spring AOP AspectJ implementation:
2. list of Annotations used is 🡺 **@Aspect, @Pointcut, @Before, @After** , **@AfterReturning, @Around, @AfterThrowing**
3. Syntax for Pointcut🡺
4.  🡺 It will be applied on all the public methods.
5. 
6. Annotation based AOP is enabled from Java 5

<aop:aspectj-autoproxy/>

1. We need the following AspectJ libraries on the classpath of your application🡺 “**aspectjrt.jar, aspectjweaver.jar, aspectj.jar, aopalliance.jar”**
2. 
3. 