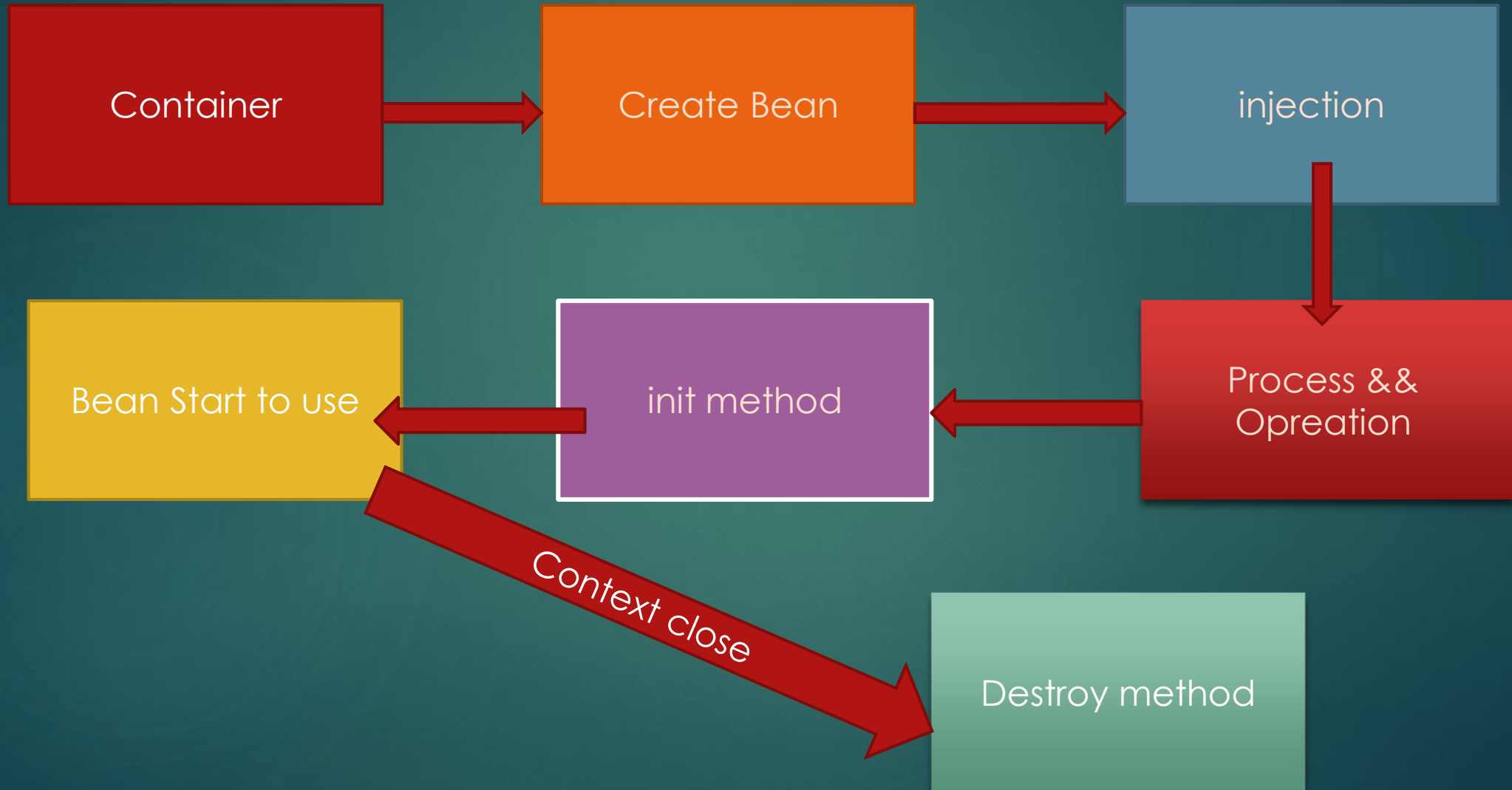


Bean Life cycle



What benefit of using Init-method

Example of you get data from database you must open connection
If in your bean function to get data so you need to add open connection in init-method

What benefit of using destroy-method

It method contain code
Example of you get data from database you must close connection at the end
so put code of close connection in destroy method

Java annotations are metadata that provide additional information about a program but do not affect its execution.

SUCH AS
`@override`

In Hibernate `@Table etc`



Java Annotations



We deal with XML Configuration
and it so hard in big project

Java annotation reduce code



Component Annotations



Spring scan Java class with Annotations

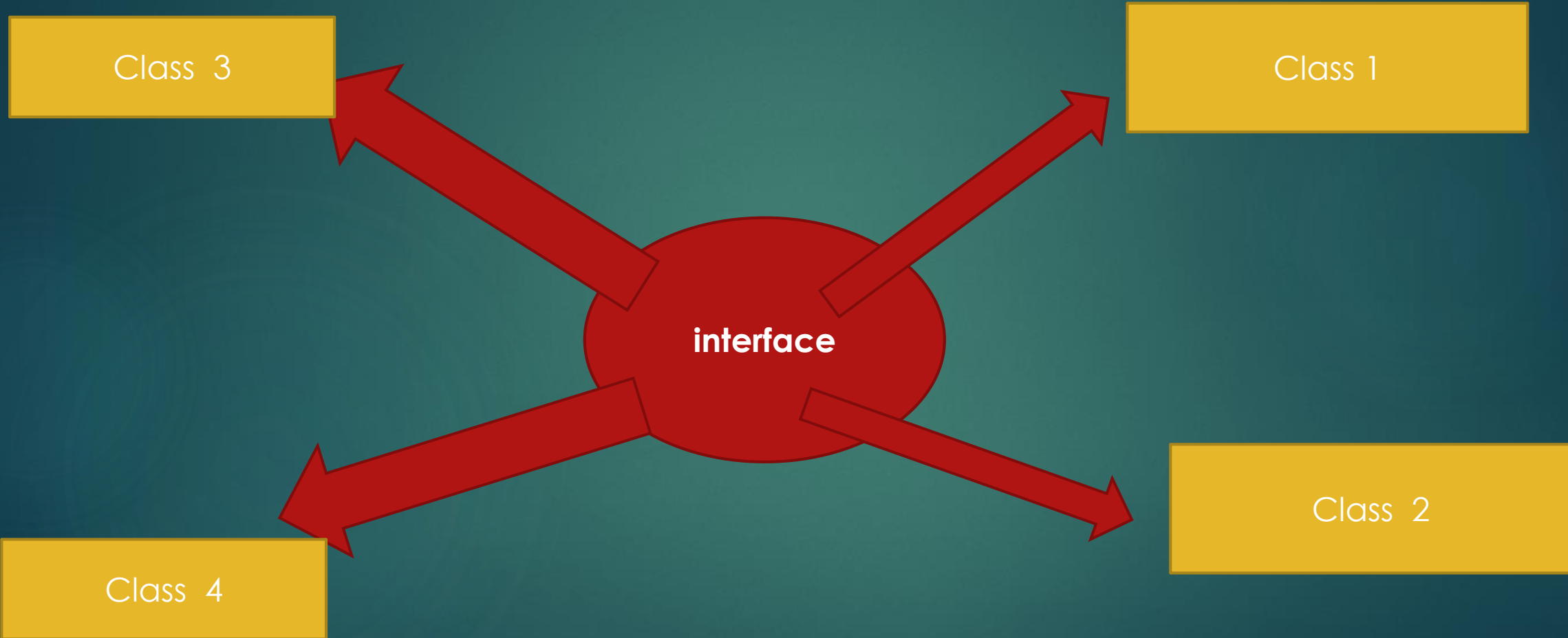
automatic register bean in containers

What is Spring AutoWiring for dependency injection

Spring Use AutoWiring to inject any thing in
beanSpring look for class and matches property

Spring automatic inject property in container

1. **Constructor Injection**
2. **Function Setter Injection**
3. **Field Injection**



Method Init-method Add custom code during bean initialization
Init-method **@PostConstruct**

Destroy-method Add custom code during bean initialization
Destroy-method **@PreDestroy**

instead of using xml container

we will create container as java code



1

```
<?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">

    <context:property-placeholder location="classpath:data.properties"/>

    <bean id="hospital" class="spring.core.HospitalService">
        <!-- <constructor-arg ref="student"></constructor-arg> -->
        <property name="repoService" ref="student"></property>
        <property name="serviceName" value="${app.service.name}"></property>
    </bean>

    <bean id="student" class="spring.core.StudentService">
```

1

```
<?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">

    <context:property-placeholder location="classpath:data.properties"/>

    <bean id="hospital" class="spring.core.HospitalService">
        <!-- <constructor-arg ref="student"></constructor-arg> -->
        <property name="repoService" ref="student"></property>
        <property name="serviceName" value="${app.service.name}"></property>
    </bean>

    <bean id="student" class="spring.core.StudentService">
```

3

```
@Configuration|  
@ComponentScan("com.spring.noxml")  
public class SpringConfig {
```