

## Spring IOC 容器中 Bean 的生命周期

**Spring IOC 容器可以管理 Bean 的生命周期**, Spring 允许在 Bean 生命周期的特定点执行定制的任务。

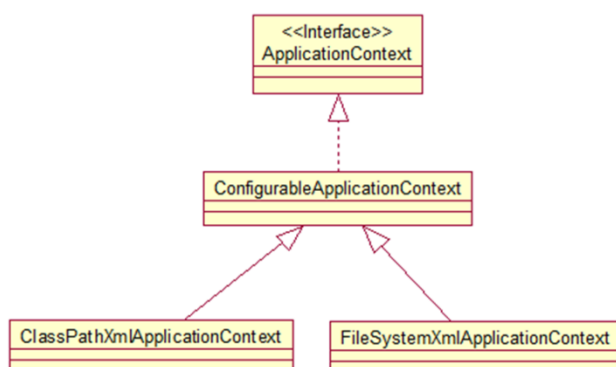
**Spring IOC 容器中 Bean 的生命周期如下:**

- ①. 通过构造器或工厂方法创建 Bean 实例: 调用构造器
- ②. 为 Bean 的属性设置值和对其他 Bean 的引用: 调用 setter
- ③. 将 Bean 实例传递给 Bean 后置处理器的 `postProcessBeforeInitialization` 方法
- ④. 调用 Bean 的初始化方法: `init-method`
- ⑤. 将 Bean 实例传递给 Bean 后置处理器的 `postProcessAfterInitialization` 方法
- ⑥. Bean 可以使用了
- ⑦. 当容器关闭时, 调用 Bean 的销毁方法: `destroy-method`。

**Bean 的初始化和销毁方法:** 可以通过 bean 节点的 `init-method` 和 `destroy-method` 来配置 Bean 的初始化方法和销毁方法:

```
<bean id="person" class="com.atguigu.spring.lifecycle.Person"
      init-method="init"
      destroy-method="destroy">
  <property name="name" value="abcd"></property>
</bean>
```

注意: `ApplicationContext` 接口中没有关闭容器的方法, 所以使用 `ApplicationContext` 接口作为 IOC 容器的引用, `destroy-method` 将不会起作用, 需要使用 `ApplicationContext` 的子接口 `ConfigurableApplicationContext`。



**详解 Bean 后置处理器:**

Bean 后置处理器: Spring 提供的特殊的 Bean

①. Bean 后置处理器允许在调用初始化方法（即：bean 节点 init-method 属性对应的方法的前后）前后对 Bean 进行额外的处理。

②. Bean 后置处理器对 IOC 容器里的所有 Bean 实例逐一处理，而非单一实例。其典型应用是：检查 Bean 属性的正确性或根据特定的标准更改 Bean 的属性。

③. 对 Bean 后置处理器而言，需要实现 BeanPostProcessor 接口

```
public class PersonPostProcessor implements BeanPostProcessor {

    /**
     * arg0: IOC 容器中 bean 的实例
     * arg1: IOC 容器中该 bean 的名字
     */
    @Override
    public Object postProcessAfterInitialization(Object arg0, String arg1)
        throws BeansException {
        if(arg0 instanceof Person){
            System.out.println("postProcessAfterInitialization");
            Person person = (Person) arg0;

            String name = person.getName();
            if(!name.equals("AAAA")){
                System.out.println("name 值必须为 AAAA!");
                person.setName("AAAA");
            }
        }

        return arg0;
    }

    @Override
    public Object postProcessBeforeInitialization(Object arg0, String arg1)
        throws BeansException {
        System.out.println("postProcessBeforeInitialization");
        return arg0;
    }

}
```

④. Bean 后置处理器需要在 IOC 容器中进行配置，但不需要指定 id 属性，Spring IOC 容器会自动的识别这是个 Bean 后置处理器，自动的使用它。

```
<bean class="com.atguigu.spring.lifecycle.PersonPostProcessor"/>
```