一、IOC

（一）什么是IOC

（1）控制反转，把对象创建和对象之间的调用过程，交给 Spring 进行管理

（2）使用 IOC 目的：为了耦合度降低

（3）做入门案例就是 IOC 实现

（二）IOC 底层原理

xml 解析、工厂模式、反射

（三）BeanFactory 接口

1、IOC 思想

IOC 思想基于 IOC 容器完成，IOC 容器底层就是对象工厂

2、Spring 提供 IOC 容器实现两种方式

（1）BeanFactory

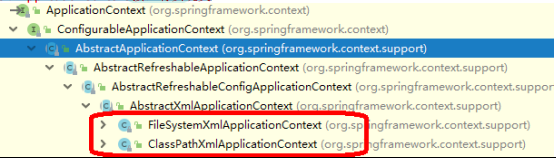
IOC 容器基本实现，是 Spring 内部的使用接口，不提供开发人员进行使用，加载配置文件时候不会创建对象，在获取对象（使用）才去创建对象

（2）ApplicationContext

BeanFactory 接口的子接口，提供更多更强大的功能，一般由开发人员进行使用

加载配置文件时候就会把在配置文件对象进行创建

3、ApplicationContext 接口有实现类



二、基于注解方式进行bean管理

（一）什么是注解

注解是代码特殊标记，格式：@注解名称(属性名称=属性值, 属性名称=属性值..)

使用注解，注解作用在类上面，方法上面，属性上面

使用注解目的：简化 xml 配置

（二）Spring 针对 Bean 管理中创建对象提供注解

@Component 常用

@Service 服务层

@Controller 控制层

@Repository DAO层

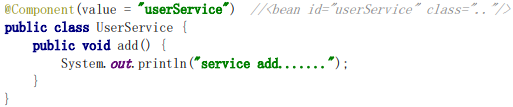
上面四个注解功能是一样的，都可以用来创建 bean 实例

（三）基于注解方式实现对象创建

1、开启组件扫描



2、创建类，在类上面添加创建对象注解



3、开启组件扫描细节配置



4、基于注解方式实现属性注入

（1）@Autowired

根据属性类型进行自动装配

（2）@Qualifier

根据名称进行注入

这个@Qualifier 注解的使用，和上面@Autowired 一起使用

（3）@Resource

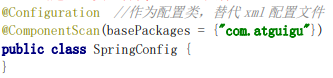
可以根据类型注入，可以根据名称注入

（4）@Value

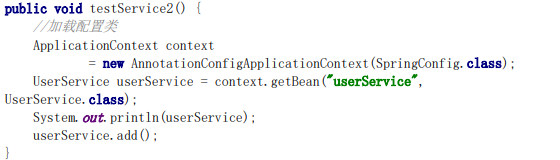
注入普通类型属性

5、完全注解开发

（1）创建配置类，替代 xml 配置文件



（2）编写测试类



三、bean 作用域

在 Spring 里面，默认情况下，bean 是单实例对象

（一）如何设置单实例还是多实例

1、scope

在 spring 配置文件 bean 标签里面有属性（scope）用于设置单实例还是多实例

2、scope 属性值

第一个值 默认值，singleton，表示是单实例对象

第二个值 prototype，表示是多实例对象

3、singleton 和 prototype 区别

singleton 单实例，prototype 多实例

设置 scope 值是 singleton 时候，加载 spring 配置文件时候就会创建单实例对象

设置 scope 值是 prototype 时候，不是在加载 spring 配置文件时候创建 对象，在调用 getBean 方法时候创建多实例对象

四、bean 生命周期

（一）生命周期

从对象创建到对象销毁的过程

（二）bean 生命周期

①通过构造器创建 bean 实例（无参数构造）

②为 bean 的属性设置值和对其他 bean 引用（调用 set 方法）

③调用 bean 的初始化的方法（需要进行配置初始化的方法）

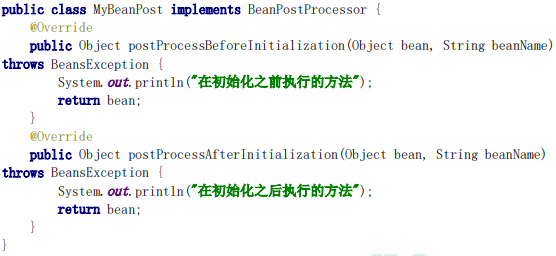
④bean 可以使用了（对象获取到了）

⑤当容器关闭时候，调用 bean 的销毁的方法（需要进行配置销毁的方法）

（三）bean 的后置处理器，bean 生命周期有七步

1、添加后置处理器

创建类，实现接口 BeanPostProcessor，创建后置处理器

****

2、最终

