**Spring优势：**

方便解耦简化开发、AOP编程支持、声明式的事务支持、方便程序测试、方便集成各种优秀的框架、降低JavaEE API的使用难度、Java源码是经典学习范例。

**程序耦合：**

降低程序间的依赖关系

在实际开发过程中做到编译时不依赖，运行时依赖。

**解耦的思路：**

第一步，使用反射创建对象，避免用new；

第二步，通过读取配置文件来获取要创建的对象全限定类名。

传统模式是主动创建，而现在是被动的，控制反转的

**JavaBean**：

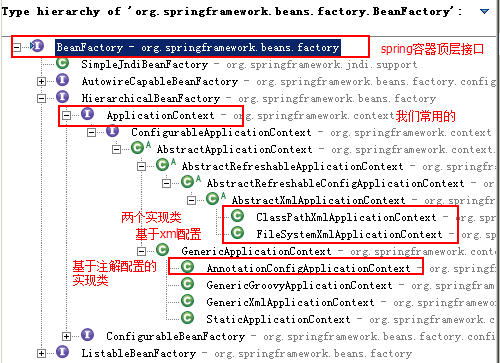
用java语言编写的可重用组件，Javabean远大于实体类。

第一个：需要一个配置文件来配置我们的service和dao

配置的内容，唯一的标识=全限定类名

第二个：通过读取配置文件中配置的内容，反射创建对象

我的配置文件可以是xml也可以是properties



**反射：**根据类名创建对象-------Class.forName()

**容器**：工厂里面的beans，hashmap，用户需要产品时，直接取出给它，不用再次反射创建

多例是多次调用时候，每个对象都被反射创建一次；而单例是用一个Map创建好，需要的时候取

**控制反转（Ioc）：**制造资源的甲方不是app本身，而是交给了乙方工厂

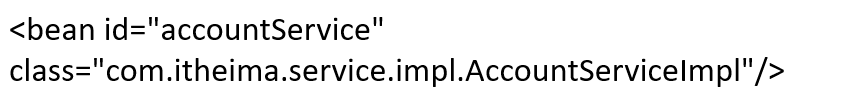
**基于XML的Ioc的概述：**

spring 使用默认无参构造函数

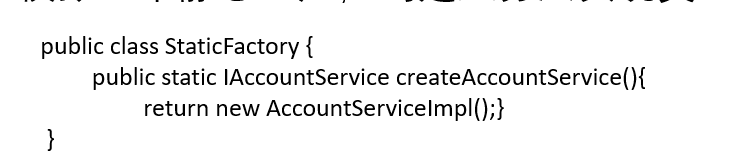
spring 管理静态工厂-使用静态工厂的方法创建对象：模拟静态工厂，使用其静态方法

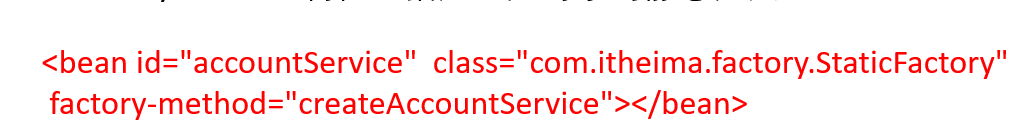
spring 管理实例工厂-使用实例工厂的方法创建对象：先指定实例工厂，再指定实例工厂的方法

1. 使用默认无参构造函数

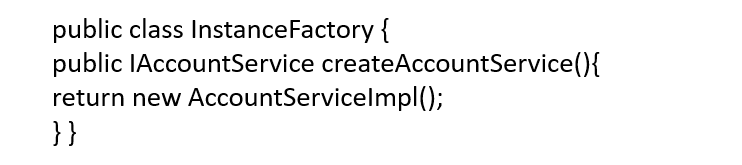


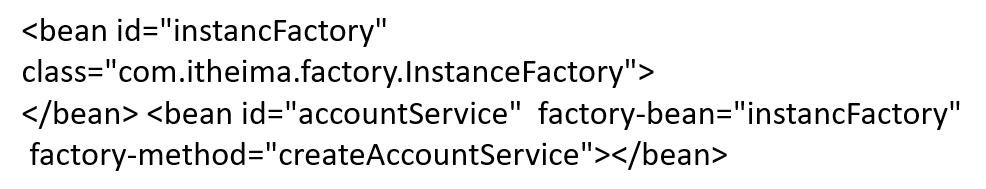
1. 使用静态工厂的方法创建对象





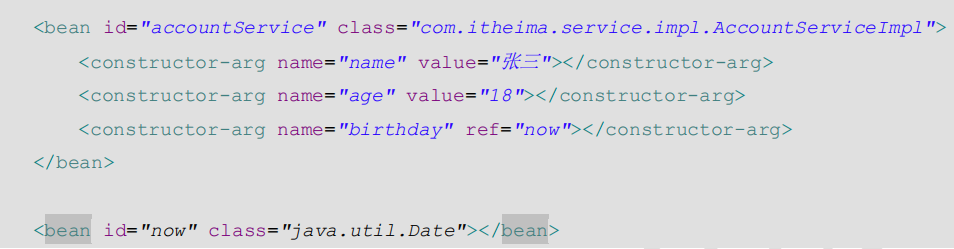
1. 使用实例工厂的方法创建对象



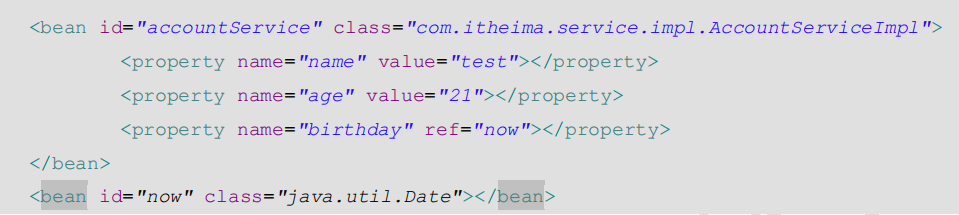


**依赖注入：**

构造函数注入



Set注入：原则是调用属性的set方法，必须有set方法



P空间注入：

导入域名解析工具，用p解析，本质上依然是set方法实现注入功能

注入集合：

Dao可以获得 Data数据库，首先建立一个连接，写Service，Client是表现层业务层。

**配置Bean：**

自己写的class直接配置，第三方class直接配置

**基于注解的方法：**

注解配置和xml配置要实现的功能是一样的，都是要降低程序间的耦合

**常用注解解析：**

@Component：把普通POJO实例化到spring容器中，相当于xml中的配置，它泛指各种组件，就是说当我们的类不属于各种归类的时候（不属于@Service、@Repository持久层、@Controller等的时候），我们就可以使用@Component来标注这个类。

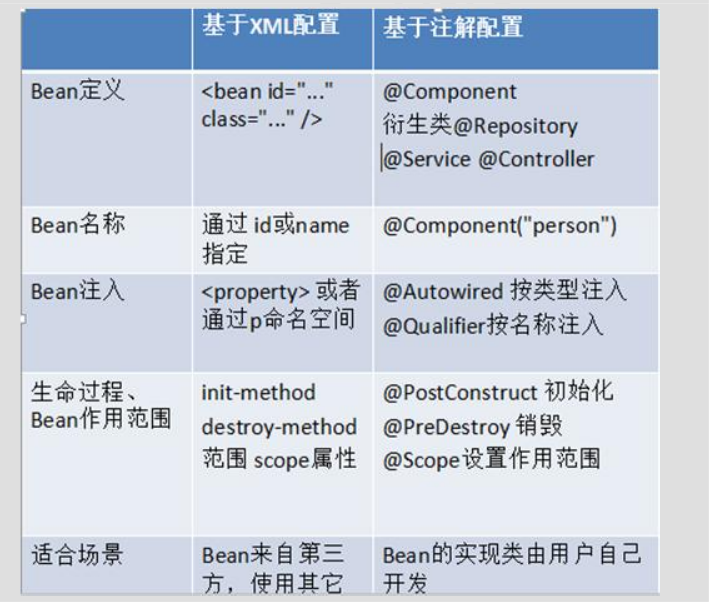
@Autowired：按照类型自动注入

@Qualifier (“”): 上面类型发生冲突时（有两个实现类）用@Qualifier的进行匹配**或者**用@Resource，@Resource是按照id注入

@Value： 基本数据类型和String类型

**IOC纯注解配置：**

把第三方工具通过Bean加到IOC容器中



**两者对比：**

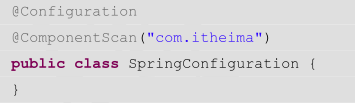
注解的优势：配置简单，维护方便（我们找到类，就相当于找到了对应的配置）。

XML 的优势：修改时，不用改源码。不涉及重新编译和部署

**纯注解配置：**

@Configuration：用于指定当前类是一个 spring 配置类，当创建容器时会从该类上加载注解。获取容器时需要使用AnnotationApplicationContext(有@Configuration 注解的类.class)。

@ComponentScan：用于指定要扫描的包，作用和在 spring 的 xml 配置文件中的<context:component-scan base-package="com.itheima"/>是一样的。

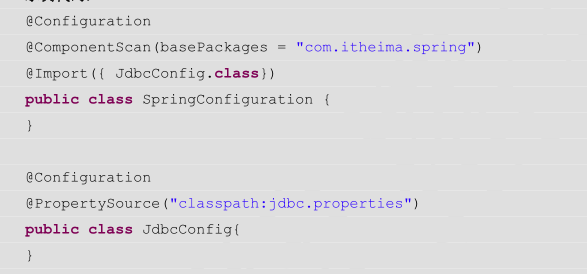


@Bean：该注解只能写在方法上，表明使用此方法创建一个对象，并且放入 spring 容器。



@PropertySource：用于加载.properties 文件中的配置。例如我们配置数据源时，可以把连接数据库的信息写到properties 配置文件中，就可以使用此注解指定 properties 配置文件的位置。

@Import：用于导入其他配置类，在引入其他配置类时，可以不用再写@Configuration 注解。



通过注解获取容器：

