1. Spring Bean的生命周期
   1. 实例化，初始化init，接收请求service，销毁destroy
   2. Spring上下文bean生命周期也类似
      1. 实例化bean：初始化bean的时候需要注入另一个尚未初始化的依赖时，容器调用createBean进行舒适化
      2. 设置对象属性（依赖注入）：实例化后的对象被封装在BeanWrapper对象中
      3. 处理Aware接口：Spring会检测对象是否出现了Aware接口，注入给bean
      4. BeanPostProcessor：如果想对Bean进行一些自定义的处理可以让bean实现beanpostprocessor接口
      5. InitializingBean与init-method：bean在Spring配置文件中配置了init-method属性。会自动调用配置的初始化方法
      6. DisposableBean：bean不需要时，会经过清理阶段
      7. destroy-method：如果这个bean的Spring配置中配置了destroy-method属性，会自动调用配置的销毁方法
2. Spring支持的集中bean作用域
   1. Spring容器中的bean可以分为五个范围
      1. Singleton
      2. Prototype
      3. Request
      4. Session
      5. Global-session
3. 使用注解前要开启自动扫描功能
   1. @configuration把一个类作为IOC容器
   2. @scope：作用域
   3. @lazy（true）表示延迟初始化
   4. @service：用于标注业务层组件
   5. @controller：用于注册控制层组件
   6. @Repository：用于标注数据访问组件
   7. @Component：泛指组件
   8. @Scope：用于指定scope作用域
   9. Postconstruct：用于指定初始化方法
   10. PreDestory:用于指定销毁方法
   11. Dependson:定义bean初始化销毁是顺序
   12. Primary:自动装配式当出现多个bean候选者
   13. Autowired:按类型装配
4. Spring框架中的单例Beans是线程安全的
   1. Spring框架并没有对单例bean进行任何多线程的封装处理，关于单例bean的线程安全和并发问题需要开发者自行去搞定，实际上大部分的Spring bean并没有可变的状态，所以在某种程度上说Spring的单例bean是线程安全的
5. Spring如何处理线程并发问题
   1. 一般情况下，在无线状态的bean才可以多线程环境共享，在Spring中，绝大部分bean都可以声明singleton作用域，因为Spring对一些bean中非线程安全状态采用threadLocal进行处理，解决线程安全问题
   2. threadLocal和线程同步机制都是为了解决多线程中相同变量的访问冲突问题，同步机制采用了时间换空间的方式，仅提供了一份变量，不用的线程在访问前需要获取锁，没获得锁的线程需要排队
6. Spring基于xml注入bean的几种方式
   1. Set方法注入
   2. 构造器注入：
   3. 静态工厂注入：
   4. 实例工厂：
7. Spring的自动装配
   1. 在Spring中，对象无需自己查找或创建与其关联的其他对象，有容器负责吧需要相互协作的对象引用赋予各个对象，使用auowire来配置自动装载模式
   2. Spring框架xml配置中共有五种个自动装配
      1. No；默认不是精心个自动装配的，通过手动设置ref属性进行封装bean
      2. Byname:通过bean的名称进行自动装配
      3. ByType:通过参数的数据类型进行自动装配
      4. Constructor:利用构造函数进行装配
      5. Autodetect:自动检测，如果有构造方法，通过construct的方法自动装配
8. @autowired和@resource之间的区别
   1. @autowired默认是按照类型装配注入，默认情况下他要求依赖必须存在
   2. @resource默认是按照名称来装配注入，找不到和名称匹配的bean会按类型
9. Spring框架中都用到了那些设计模式
   1. 工厂模式：beanfactory就是简单工厂模式的体现，用来创建对象的实例
   2. 单例模式：bean默认为单例模式
   3. 代理模式：Spring的aop功能用到了jdk动态代理和cglib字节码
   4. 模板方法：用来解决代码重复的问题
   5. 观察者模式：定义对象键一种一对多的依赖关系
10. Spring事务的实现方式和实现原理
    1. Spring事务的本质就是数据库对事物的支持，没有数据库的事务支持，Spring是无法提供事务功能的，真正的数据层的事务提交和回滚是通过binlog或者redolog实
    2. Spring事务的种类

Spring支持变成事务管理和生命事务管理的两种方式

编程式事务管理使用transactiontemplate

生命式事务管理建立在Aop之上，本质是通过aop功能

* 1. Spring的事务传播行为
     1. Propagation\_required
     2. Propagation\_supports
     3. Propagation\_mandatory
     4. Propagation\_requires\_new
     5. Propagation\_not\_supported
     6. Propagation\_never
     7. Propagation\_nested