Spring

1. 什么是Spring？

。Spring就是把每个bean(实体类/对象)与bean的关系全部交给第三方容器进行管理，这个容器就是Spring，对整个对象的生命周期进行管理的。

。Spring的核心是IOC(控制反转：创建对象)，DI(依赖注入：做注入)，AOP(面向编程)。

。Spring可以整合很多框架

（把继承对象交给Spring容器进行管理）

（我们用的Spring默认是单例的，单例是共享数据是线程不安全）

（int的包装类是Integer，int不允许为null,默认值为0，Integer可以为null）

（spring中使用饿汉式，会考虑线程安全问题）

SpringIOC

1. SpringIOC原理是什么？

。什么是SpringIOC：控制反转，把bean交给spring容器进行管理赋值

。1.解析XML配置，获取bean class地址，2.使用Java反射机制 进行实例化对象。3.返回对象

1. Spring中bean的id重复了会怎么样？

。会报错，分两种情况：在xml中bean的id不允许重复，如果是注解和xml一起的话不会报错

1. Spring四种生命周期

1.singleton(单例)默认（JVM只能运行存在一次。。单例分为懒汉式和饿汉式。使用单例注意事项：线程安全。）

2.Prototype(原型)多例：调用getbean方法才能初始化

3.request ：http请求request作用域里面使用

4.session ：http请求session作用域里面使用

五．Spring的核心IOC 和DI的区别

IOC：控制反转，负责创建对象：

创建对象的几种方式：无参数构造器，（默认）

带参构造函数

工厂创建对象（静态，非静态）

DI：依赖注入，解决对象之间的依赖关系

依赖注入如何给对象赋值？

1.通过构造函数

2.通过set方法给属性注入值（原理：属性必须要有一个set方法。通过set反射进行注入，不写set会报错）

3.P名称空间（p标签：通过反射机制必须要生成一个set方法进行赋值）

4.自动装配（了解）

5.注解

（Spring中jar包必须要有Spring\_core,Spring-context）

（类的完整路径用到的是反射）

SpringAOP

1. 什么是SpringAOP:

。面向切面编程 （是一种思想）

。应用场景：权限控制，事务管理，日志打印，性能统计

。什么地方使用AOP:代码重复，在不同的方法中要做同样的事

（事物的原理是AOP，作用是保证数据的一致性）

。SpringAOP 代理设计模式（代理模式的作用是提供目标对象进行访问的一个方式，类似中介）

。动态代理：不需要生成代理类，分为JDK动态代理（反射机制），cglib动态代理（字节码）。区别是JDK使用反射机制，cglib使用字节码（asm）

。静态代理：需要生成代理类

1. SpringAOP编程的作用是将代码分离

。功能是让关注点代码与业务代码分离！

。关注点：重复的代码就叫做关注点

。切面：关注点形成的类，就叫做切面（类）

.面向切面编程，就是指对很多功能都有的重复代码抽取，再在运行的时候网业务方法上动态植入“切面类代码”（抽出重复代码）

@Aspect(定义切面)

@Compontent(注入Spring容器 )

。切入点：执行目标对象方法，动态植入切面代码

.可以通过切入点表达式，制定拦截哪些类的哪些方法，给指定的类在运行的时候植入切面代码类

1. 注解

。前置通知（@before）

。后置通知（@After）

。异常通知（@AfterThrowing）

。运行通知（@AfterReturning）

。环绕通知（@Around）

AOP: Aspect Oriented Programming 面向切面编程。  
　　面向切面编程(也叫面向方面)：Aspect Oriented Programming(AOP),是目前软件开发中的一个热点。利用AOP可以对业务逻辑的各个部分进行隔离，从而使得业务逻辑各部分之间的耦合度降低，提高程序的可重用性，同时提高了开发的效率。  
　　AOP是OOP的延续，是（Aspect Oriented Programming）的缩写，意思是面向切面（方面）编程。

（MVC是一种架构思想，不是设计模式）

Spring事物与传播行为

1. 什么是事物？

保证数据的一致性

1. 数据的特性
2. 原子性：事物包含的所有操作要么全部成功，要么全部失败回滚。成功就必须完全应用到数据库，失败则不能对数据库有任何影响（不可分割的，要么执行要么不执行）
3. 一致性：是指事物必须是数据库从一个一致性状态变换到另一个一致性状态，也就是说一个事务执行之前和执行之后都必须处于一致性状态
4. 隔离性：是指多个用户并发访问数据库时，比如操作同一张表时，数据库为每一个用户开启的事务，不能被其他事务的操作所干扰，多个并发事务之间要相互隔离（互不影响）
5. 持久性：持久性是指一个事务一旦被提交了，那么对数据库中的数据的改变就是永久性的，即便是在 系统遇到故障的情况下也不会丢失提交事务的操作。

（事物放到service业务逻辑层）