1. 什么是spring？使用spring框架的好处？

Spring 是java企业应用级的开发框架，它的目标是简化java的开发。

使用spring框架的好处：

1. spring是轻量级的，每个版本在2MB
2. 控制反转：spring通过控制反转实现了松散耦合度.

控制反转的底层实现原理？

在控制反转出来之前，业务的底层代码对象与对象之间有着复杂的依赖关系。控制反转的理念就是把复杂的系统分解为相互合作的对象，这些对象通过封装后，内部实现是对外部是透明的，从而降低了解决问题的复杂度。其借助于“第三方”解决了实现了具有依赖关系的解耦。所谓依赖注入，就是由IoC容器在运行期间，动态地将某种依赖关系注入到对象之中。

使用ioc的好处每个单元都是独立的，非常方便测试；每个开发团队只需要顾及自己所涉及的那部分代码。

AOP原理？

主要应用于两个方面：

1. 面向切面编程提供声明式事务
2. Spring支持用户自定义的切面
3. 将日志记录，性能统计，安全控制，事务处理，异常处理等代码从业务逻辑代码中划分出来，通过对这些行为的分离，我们希望可以将它们独立到非指导业务逻辑的方法中，进而改变这些行为的时候不影响业务逻辑的代码

AOP组成：

1，连接点（JoinPooint）：程序执行的某个特定位置，如类的开始初始化前，类的初始化后，类的某个方法调用前/后，方法抛出异常后。一个类或者一段代码拥有一些边界性质的特定点，这就是连接点。Spring仅支持方法的连接点，即仅能在方法调用前，方法调用后，方法抛出异常时及方法调用前后这些程序执行点织入增强；

2，切点（Pointcut）：每个程序类都拥有多个连接点，在多个连接点中定位我们想要的连接点。AOP通过切点定位连接点，切点和连接点不是一对一的关系，一个切点可以匹配多个连接点。。

3，增强（Advice）：增强是织入到目标类连接点上的一段程序代码，Spring中增强除了用于描述一段程序代码外，还拥有另一个和连接点相关的信息，执行点的方位。Spring 提供的接口都是带方位名的：BeforeAdvice、AfterReturningAdvice，等

4，目标对象（Target）：增强逻辑的织入目标类

5，引介（Introduction）：引介是特殊的增强，他为类添加一些属性和方法。

6，织入（Weaving）：将增强添加到目标类的具体链接点的过程。

7，代理（Proxy）：

8，切面（Aspect）：切面由切点和增强组成，他既包括横切逻辑的定义，也包括连接点的定义。

代理模式提供了一种在不改变原来对象的基础上，通过代理对象来访问目标对象的机制。

1. 静态代理

在使用的时候，需要定义接口或者父类。被代理的对象与代理的对象一同实现相同的接口或继承相同的父类。然后通过调用代理对象的方法去调用目标对象的方法

1. 动态代理

Jdk代理和cglib代理

1. Jdk动态代理

接口代理。

Jdk动态代理通过接口和类加载器创建代理类。具体而言是通过**调用Proxy.newProxyInstance()方法创建代理对象。**

第一个参数指定classLoader，第二个参数指定代理对象要实现的接口，第三个对象那个是InvocationHandler类型。来看下InvocationHandler：

一个代理对象和一个InvocationHandler绑定，当执行代理对象的方法时，就会去执行InvocationHandler的invoke(Object proxy, Method method, Object[] args)方法，在该方法中你可以选择相应的处理或者不执行代理对象的method方法。

1. Cglib代理

Cglib代理利用拦截器加上反射机制生成一个实现代理接口的匿名类。Cglib利用了asm开源包，对代理对象类的class文件加载进来，通过修改字节码生成子类代理。（cglib生成代理类的二进制字节码文件；加载二进制字节码，生成class对象；通过反射机制获得实例构造，并创建代理对象）

Spring框架最核心的两个包是org.springframework.beans.factory（该包中的主要接口是BeanFactory）和org.springframework.context包（该包中的主要接口是ApplicationFactory）。

Spring ioc的基础组件：beans，配置文件beans.xml和applicationContext.xml。BeanFactory相关接口及其相关类；applicationContext相关接口及其相关类。

Beanfactory采用了工厂设计模式，即bean容器模式，负责读取bean的配置文件，管理对象的生成，加载，维护bean对象之间的依赖关系。rg.springframework.beans.factory.xml.XmlBeanFactory是BeanFactory常用的实现类。其根据配置文件中的定义装载Bean。要创建XmlBeanFactory，需要传递一个FileInputStream对象，该对象把XML文件提供给工厂。

— BeanFactory的常用方法如下：

getBean(String name)：该方法可根据Bean的id生成该Bean的对象。

getBean(String name,Class requiredType)：该方法可根据Bean的id和相应类生成该Bean的对象。

ApplicationContext接口及其相关类：

— 作为一个应用程序框架，只提供Bean容器管理的功能是不够的。若要利用Spring3所提供的一些高级容器功能，则可以使用ApplicationContext接口，该接口是提供高级功能的容器。

— ApplicationContext的基本功能与BeanFactory很相似，但它还有以下功能：

提供访问资源文件的更方便的方法。

支持国际化消息。

提供文字消息解析的方法。

可以发布事件，对事件感兴趣的Bean可以接收到这些事件。

— ApplicationContext接口的常用实现类有3个：

FileSystemXmlApplicationContext：从文件系统中的XML文件加载上下文中定义的信息。

ClassPathXmlApplicationContext：从类路径中的XML文件加载上下文中定义的信息，把上下文定义的文件当成类路径资源。

XmlWebApplicationContext：从Web系统中的XML文件加载上下文中定义的信息。

Spring ioc的生命周期？

通过构造器或者工厂方法创建 bean 实例。

为 bean 的属性赋值和对其他 bean 的引用。

调用 bean 的初始化方法。

bean 初始成功，可以使用。

容器关闭时 , 调用 bean 的销毁方法。

Spring bean的作用域？

1. singleton

此时spring的ioc容器只有存在一个共享的bean实例

1. prototype

每次对该bean的请求都会创建一个新的bean实例

1. request

在一次http请求中，一个bean对应一个实例

1. global
2. Session

**Spring的事务**

Spring分为编程式事务和生命式事务

声明式事务是建立在aop之上的；其本质是对方法前后拦截；然后在目标方法前加入一个事务；在执行完目标方法后根据执行情况提交或者回滚事务。

**Spring的事务传播特性？**

1. 表示该方法必须运行在一个事务中。如果当前没有事务正在发生，将抛出一个异常
2. 表示如果当前正有一个事务在进行中，则该方法应当运行在一个嵌套式事务中。被嵌套的事务可以独立于封装事务进行提交或回滚。如果封装事务不存在，行为就像PROPAGATION\_REQUIRES一样。
3. 表示当前的方法不应该在一个事务中运行。如果一个事务正在进行，则会抛出一个异常。
4. 表示当前的方法不应该在一个事务中运行。如果一个事务正在进行，则会挂起

Spring如何处理线程并发问题？

使用ThreadLocal线程本地变量，threadlocal为变量在每个线程中都创建了一个副本。每个线程可以访问自己内部的副本变量，各线程之间互不干扰。

Theadlocal:方法：get和set。

Set：获取当前对象Threadlocal;如果map不为空，将threadlocal和新的value副本放到map中；否则先进行初始化创建。

每个thread对象都有一个threadlocalmap，当创建一个threadlcoalmap时候就会将threadlocal对象添加到map中，主键就是threadlocal；值可以是任意类型。 Threadlocalmap解决hash冲突使用线性探测，简单的步长加1或者减1。

Threadlocalmap中，key是弱引用，value是强引用。所以value不会被回收，最终导致内存泄漏。当使用完get和set方法后，一定要使用remove（）方法。