二十一、AOP执行顺序：

springboot1：
正常情况下：@Before前置通知----->@After后置通知----->@AfterRunning正常返回
异常情况下：@Before前置通知----->@After后置通知----->@AfterThrowing方法异常
springboot2：
正常：我是环绕通知之前AAA
**@Before我是前置通知
===>CalcServiceImpl被调用，计算结果为：5
**@AfterReturning我是返回后通知
**@After我是后置通知
我是环绕通知之后BBB
异常：我是环绕通知之前AAA
**@Before我是前置通知
@AfterThrowing我是异常通知
**@After我是后置通知
java.lang.ArithmeticException: / by zero

二十二、什么是循环依赖？如何解决？

多个bean之间相互依赖，形成了一个闭环。比如：A依赖于B、B依赖于C、C依赖于A。会报BeanCurrentlyInCreationException

```
AB循环依赖问题只要A的注入方式是set注入或bean本身是singleton ，就不会有循环依赖问题。(构造注入因为实例还没构造，无法提前曝光)
spring内部通过三级缓存来解决循环依赖问题。
```

二十三、spring内部如何通过三级缓存来解决循环依赖问题？

DefaultSingletonBeanRegistry
只有单例的bean会通过三级缓存提前暴露来解决循环依赖的问题，而非单例的bean，每次从容器中获取都是一个新的对象，都会重新创建，所以非单例的bean是没有缓存的，不会将其放到三级缓存中。

```
第一级缓存（也叫单例池）singletonObjects：存放已经经历了完整生命周期的Bean对象。（成品）
第二级缓存：earlySingletonObjects，存放早期暴露出来的Bean对象，Bean的生命周期未结束（属性还未填充完。 （半成品）
第三级缓存：Map<String, ObjectFactory<?>> singletonFactories，存放可以生成Bean的工厂。 （准备生产）

A / B两对象在三级缓存中的迁移说明：
    A创建过程中需要B，于是A将自己放到三级缓里面，去实例化B。
    B实例化的时候发现需要A，于是B先查一级缓存，没有，再查二级缓存，还是没有，再查三级缓存，找到了A然后把三级缓存里面的这个A放到二级缓存里面，并删除三级缓存里面的A。
    B顺利初始化完毕，将自己放到一级缓存里面（此时B里面的A依然是创建中状态），然后回来接着创建A，此时B已经创建结束，直接从一级缓存里面拿到B，然后完成创建，并将A自己放到一级缓存里面。
主要方法：
    getSingleton()、doCreatBean()、populateBean()、addSingleton()
```

二十四、Redis基本类型及使用场景？

String：
商品编号、订单号采用INCR命令生成
点赞
Hash：
购物车早期，当前小中厂可用
List:
微信文章订阅公众号,粉丝列表
Set：
微信抽奖小程序、微信朋友圈点赞、共同关注的人、猜你喜欢
ZSet：
根据商品销售量排序显示

二十五、分布式锁

多个服务间 + 保证同一时刻内 + 同一用户只能有一个请求（防止关键业务出现数据冲突和并发错误）

```
卖票程序：1->2加锁synchronized
        2->3加分布式锁 RedisTemplate
        3->4必须finally释放锁
        4->5防止机器宕机，需要给redis的key设置过期时间
        5->6加锁和设置expire必须要是原子操作
        6->7线程可能会误删他人锁，一定要判断是自己的锁
        7->8判断自己锁和删除锁不是原子操作。解锁过程可用lua脚本、Redis事务、Redisson
        8->9风险：Redis过期时间大于业务执行、Redis分布式锁如何续期？
                Redis - AP -redis异步复制造成的锁丢失，比如：主节点没来的及把刚刚set进来
这条数据给从节点，就挂了。
                (ZooKeeper - CP)
                (CAP
                    C：Consistency（强一致性）
                    A：Availability（可用性）
                    P：Partition tolerance（分区容错性）)
                解决：Redis集群环境下，我们自己写的也不OK，直接上RedLock之Redisson落地实
现。
        9->10健壮：解锁前判断是否加锁，并且锁是自己持有的
```

二十六、Redis内存满了怎么办？Redis默认内存多少？在哪里查看？如何设置修改？

查看Redis最大占用内存
配置文件redis.conf的maxmemory参数，maxmemory是bytes字节类型，注意转换。
redis默认内存多少可以用？
如果不设置最大内存大小或者设置最大内存大小为0，在64位操作系统下不限制内存大小，在32位操作系统下最多使用3GB内存
一般生产上你如何配置？
一般推荐Redis设置内存为最大物理内存的四分之三。
如何修改redis内存设置
1修改配置文件redis.conf的maxmemory参数，如：maxmemory 104857600（100MB）
2通过命令修改
config set maxmemory 1024
config get maxmemory
什么命令查看redis内存使用情况?
info memory

二十七、如果Redis内存使用超出了设置的最大值会怎样?

(error) OOM command not allowed when used memory > 'maxmemory'.

二十八、Redis内存淘汰策略，如何配置，修改？

noeviction：不会驱逐任何key（不要用）
volatile-lfu：对所有设置了过期时间的key使用LFU算法进行删除
volatile-lru：对所有设置了过期时间的key使用LRU算法进行删除
volatile-random：对所有设置了过期时间的key随机删除
volatile-ttl：删除马上要过期的key
allkeys-lfu：对所有key使用LFU算法进行删除
allkeys-lru：对所有key使用LRU算法进行删除（最常用）
allkeys-random：对所有key随机删除（不要用）

```
命令：
    config set maxmemory-policy noeviction
    config get maxmemory-policy
配置文件 - 配置文件redis.conf的maxmemory-policy参数
```

二十九、过滤器和拦截器的区别？

过滤器：
　　是在java web中，你传入的request,response提前过滤掉一些信息，然后再传入servlet进行业务逻辑，
　　比如过滤掉非法url（不是login.do的地址请求，如果用户没有登陆都过滤掉）,或者在传入servlet前统一设置字符集，或者去除掉一些非法字符
拦截器：
　　是在面向切面编程的就是在你的service或者一个方法，前调用一个方法，或者在方法后调用一个方法比如动态代理就是拦截器的简单实现，

```
拦截器与过滤器的区别 ：
    1.拦截器是基于java的反射机制的，而过滤器是基于函数回调。
    2.拦截器不依赖与servlet容器，过滤器依赖与servlet容器。
    3.拦截器只能对action请求起作用，而过滤器则可以对几乎所有的请求起作用。
    4.拦截器可以访问action上下文、值栈里的对象，而过滤器不能访问。
    5.在action的生命周期中，拦截器可以多次被调用，而过滤器只能在容器初始化时被调用一次
```

三十、OSI ， TCP/IP ，五层协议的体系结构？

物理层：激活、维持、关闭通信端点之间的机械特性、电气特性、功能特性以及过程特性。
　　该层为上层协议提供了一个传输数据的物理媒体。

```
数据链路层 ：数据链路层在不可靠的物理介质上提供可靠的传输。
    该层的作用包括：物理地址寻址、数据的成帧、流量控制、数据的检错、重发等。

网络层 ：网络层负责对子网间的数据包进行路由选择。
    此外，网络层还可以实现拥塞控制、网际互连等功能。

传输层 ：第一个端到端，即主机到主机的层次。传输层负责将上层数据分段并提供端到端的、可靠的或不可靠的传输。
    此外，传输层还要处理端到端的差错控制和流量控制问题。
```

会话层 ：会话层管理主机之间的会话进程，即负责建立、管理、终止进程之间的会话。
　　会话层还利用在数据中插入校验点来实现数据的同步。

表示层 ：表示层对上层数据或信息进行变换以保证一个主机应用层信息可以被另一个主机的应用程序理解。
　　表示层的数据转换包括数据的加密、压缩、格式转换等。

应用层 ：为操作系统或网络应用程序提供访问网络服务的接口。

三十一、ARP 协议的工作原理

首先，每台主机都会在自己的ARP缓冲区中建立一个 ARP列表，以表示IP地址和MAC地址的对应关系。

当源主机需要将一个数据包要发送到目的主机时，会首先检查自己 ARP列表中是否存在该 IP地址对应的MAC地址，

如果有，就直接将数据包发送到这个MAC地址；如果没有，就向本地网段发起一个ARP请求的广播包，查询此目的主机对应的MAC地址。

此ARP请求数据包里包括源主机的IP地址、硬件地址、以及目的主机的IP地址。

网络中所有的主机收到这个ARP请求后，会检查数据包中的目的IP是否和自己的IP地址一致。

如果不相同就忽略此数据包；如果相同，该主机首先将发送端的MAC地址和IP地址添加到自己的ARP列表中，如果ARP表中已经存在该IP的信息，则将其覆盖，

然后给源主机发送一个 ARP响应数据包，告诉对方自己是它需要查找的MAC地址；

源主机收到这个ARP响应数据包后，将得到的目的主机的IP地址和MAC地址添加到自己的ARP列表中，并利用此信息开始数据的传输。

如果源主机一直没有收到ARP响应数据包，表示ARP查询失败。

三十二、常见的路由选择协议，以及它们的区别？

常见的路由选择协议有：RIP协议、OSPF协议。

RIP协议 ：内部网关协议，底层是贝尔曼福特算法，它选择路由的度量标准（metric)是跳数，最大跳数是15跳，如果大于15跳，它就会丢弃数据包。

OSPF协议 ：内部网关协议，底层是迪杰斯特拉算法，是链路状态路由选择协议，它选择路由的度量标准是带宽，延迟。

BGP协议：外部网关协议。

三十三、在浏览器中输入 www.baidu.com 后执行的全部过程？

现在假设如果我们在客户端（客户端）浏览器中输入http://www.baidu.com,而baidu.com为要访问的服务器（服务器），下面详细分析客户端为了访问服务器而执行的一系列关于协议的操作：

1）客户端浏览器通过DNS解析到www.baidu.com的IP地址220.181.27.48，通过这个IP地址找到客户端到服务器的路径。

客户端浏览器发起一个HTTP会话到220.161.27.48，然后通过TCP进行封装数据包，输入到网络层。

2）在客户端的传输层，把HTTP会话请求分成报文段，添加源和目的端口，

如服务器使用80端口监听客户端的请求，客户端由系统随机选择一个端口如5000，与服务器进行交换，服务器把相应的请求返回给客户端的5000端口。然后使用IP层的IP地址查找目的端。

3）客户端的网络层不用关系应用层或者传输层的东西，主要做的是通过查找路由表确定如何到达服务器，期间可能经过多个路由器，

这些都是由路由器来完成的工作，我不作过多的描述，无非就是通过查找路由表决定通过那个路径到达服务器。

4）客户端的链路层，包通过链路层发送到路由器，通过邻居协议查找给定IP地址的MAC地址，

然后发送ARP请求查找目的地址，如果得到回应后就可以使用ARP的请求应答交换的IP数据包现在就可以传输了，然后发送IP数据包到达服务器的地址。

三十四、HTTP 中， POST 与 GET 的区别

(1)Get是从服务器上获取数据，Post是向服务器传送数据。
(2)Get是把参数数据队列加到提交表单的Action属性所指向的URL中，值和表单内各个字段一一对应，在URL中科院看到。
(3)Get传送的数据量小，不能大于2KB；post传送的数据量较大，一般被默认为不受限制。
(4)根据HTTP规范，GET用于信息获取，而且应该是安全的和幂等的。

I.所谓 安全的 意味着该操作用于获取信息而非修改信息。换句话说，GET 请求一般不应产生副作用。就是说，它仅仅是获取资源信息，就像数据库查询一样，不会修改，增加数据，不会影响资源的状态。

II. 幂等 的意味着对同一URL的多个请求应该返回同样的结果。

三十五、TCP/IP 中，每一层对应的协议

网络层 ： IP协议、ICMP协议、ARP协议、RARP协议。

传输层 ： UDP协议、TCP协议。

应用层 ： FTP（文件传送协议）、Telenet（远程登录协议）、DNS（域名解析协议）、SMTP（邮件传送协议），
POP3协议（邮局协议），HTTP协议。

三十六、TCP 对应的协议和 UDP 对应的协议

TCP对应的协议：

（1） FTP ：定义了文件传输协议，使用21端口。常说某某计算机开了FTP服务便是启动了文件传输服务。下载文件，上传主页，都要用到FTP服务。

（2） Telnet ：它是一种用于远程登陆的端口，用户可以以自己的身份远程连接到计算机上，通过这种端口可以提供一种基于DOS模式下的通信服务。

如以前的BBS是-纯字符界面的，支持BBS的服务器将23端口打开，对外提供服务。

（3） SMTP ：定义了简单邮件传送协议，现在很多邮件服务器都用的是这个协议，用于发送邮件。

如常见的免费邮件服务中用的就是这个邮件服务端口，所以在电子邮件设置-中常看到有这么SMTP端口设置这个栏，服务器开放的是25号端口。

（4） POP3 ：它是和SMTP对应，POP3用于接收邮件。通常情况下，POP3协议所用的是110端口。
也是说，只要你有相应的使用POP3协议的程序（例如Fo-xmail或Outlook），就可以不以Web方式登陆进邮箱界面，直接用邮件程序就可以收到邮件（如是163邮箱就没有必要先进入网易网站，再进入自己的邮-箱来收信）。

（5）HTTP协议： 是从 Web 服务器传输超文本到本地浏览器的传送协议。

UDP对应的协议：

（1） DNS ：用于域名解析服务，将域名地址转换为IP地址。DNS用的是53号端口。

（2） SNMP ：简单网络管理协议，使用161号端口，是用来管理网络设备的。由于网络设备很多，无连接的服务就体现出其优势。

（3） TFTP (Trival File Transfer Protocal)，简单文件传输协议，该协议在熟知端口69上使用UDP服务。

三十七、NAT 协议、 DHCP 协议、 DNS 协议的作用？

NAT协议 ：网络地址转换(NAT,Network AddressTranslation)属接入广域网(WAN)技术，
是一种将私有（保留）地址转化为合法IP地址的转换技术，它被广泛应用于各种类型Internet接入方式和各种类型的网络中。
原因很简单，NAT不仅完美地解决了IP地址不足的问题，而且还能够有效地避免来自网络外部的攻击，隐藏并保护网络内部的计算机。

DHCP协议 ：动态主机设置协议（Dynamic Host ConfigurationProtocol，DHCP）
是一个局域网的网络协议，使用UDP协议工作，主要有两个用途：给内部网络或网络服务供应商自动分配IP地址，给用户或者内部网络管理员作为对所有计算机作中央管理的手段。

DNS协议 ：DNS 是域名系统 (Domain Name System) 的缩写，是因特网的一项核心服务，
它作为可以将域名和IP地址相互映射的一个分布式数据库，能够使人更方便的访问互联网，而不用去记住能够被机器直接读取的IP数串。

三十八、happen-before 原则

程序顺序规则：一个线程内执行的每个操作，都保证 happen-before 后面的操作，这样就保证了程序顺序规则，
volatile 变量规则：对于 volatile 变量，对他的写操作，保证 happen-before 在随后对改变量的读取操作。
监视器锁规则：对于一个锁的解锁操作，保证 happen-before 加锁操作。
线程启动 happen-before 规则：Thread 对象的 start() 方法先行于此线程的每一个动作
线程中断 happen-before 规则：对线程 interrupt() 方法的调用先行发生于被中断线程的代码检测到中断事件的发生。
线程终结规则：假定线程A在执行的过程中，通过制定ThreadB.join()等待线程B终止，那么线程B在终止之前对共享变量的修改在线程A等待返回后可见。
对象终结规则，一个对象的初始化完成happen-before 于 finalizer() 方法的开始
传递性：happen-before 存在传递性，a happen-before b ,b happen-before c ,那么 a happen-before c 。
happen-before 保障了顺序执行，也包括了内存读写的操作顺序。

三十九、HTTP的常见状态码有哪些，代表什么含义？

常见状态码：

200 OK：正常返回信息

400 Bad Request：客户端请求有语法错误，不能被服务器所理解

403 Forbidden：服务器收到请求，但是拒绝提供服务

404 Not Found：请求资源不存在，输入了错误的URL

500 Internal Server Error：服务器发生不可预期错误

503 Server Unavailable：服务器当前不能处理客户端的请求，一段时间后可能恢复正常

四十、什么是加密？什么是数字签名？什么是数字证书？区别？

加密：用加密算法将加密对象处理，用公钥加密，私钥解密。保证数据不被别人看到。

数字签名：用私钥加密，公钥解密。保证数据是我自己发的，有任何篡改都能被识别、

数字证书：对称加密中，双方使用公钥进行解密。虽然数字签名可以保证数据不被替换，

但是数据是由公钥加密的，如果公钥也被替换，则仍然可以伪造数据，因为用户不知道对方提供的公钥其实是假的。

所以为了保证发送方的公钥是真的，CA 证书机构会负责颁发一个证书，里面的公钥保证是真的，

用户请求服务器时，服务器将证书发给用户，这个证书是经由系统内置证书的备案的。