

# 1、锁可以锁在哪里？

Java 为程序加锁的方式主要有两种：synchronized 与 Lock。

1. synchronized 可以修饰的作用域如下：

- 非静态方法（加的锁为对象锁）；
- 静态方法（加的锁为类锁）；
- 代码块（对象锁与类锁均可）；

2. Lock 采用 lock()对代码加锁，unlock()进行解锁

参考文章：<https://blog.csdn.net/yx0628/article/details/79086511>

# 2、怎么利用反射获取类中的对象？

1. 获取 Class 对象

2. 通过 Class 对象获取构造方法

3. 通过构造方法调用 newInstance()方法创建对象实例

参 考 文 章 :  
<https://baijiahao.baidu.com/s?id=1619748187138646880&wfr=spider&for=pc>

# 3、HTTP 和 https 区别？

1. HTTP 明文传输，数据都是未加密的，安全性较差，HTTPS（SSL+HTTP）数据传输过程是加密的，安全性较好。

2. 使用 HTTPS 协议需要到 CA（Certificate Authority，数字证书认证机构）申请证书，一般免费证书较少，因而需要一定费用。

3. HTTP 页面响应速度比 HTTPS 快，主要是因为 HTTP 使用 TCP 三次握手建立连接，客户端和服务端需要交换 3 个包，而 HTTPS 除了 TCP 的三个包，还要加上 ssl 握手需要的 9 个包，所以一共是 12 个包。

4. http 和 https 使用的是完全不同的连接方式，用的端口也不一样，前者是 80，后者是 443。

5. HTTPS 其实就是建构在 SSL/TLS 之上的 HTTP 协议，所以，要比较 HTTPS 比 HTTP 要更耗费服务器资源。

参考文章：<https://www.runoob.com/w3cnote/http-vs-https.html>

# 4、谈谈你对 TCP / IP 协议的了解

1. TCP/IP 是一个协议簇，是一个开放的协议标准，所有人都可以免费使用，并且是独立于硬件和操作系统的。

2. TCP/IP 协议是不区分网络硬件的，它在局域网，广域网和互联网中都被广泛使用。

3. TCP/IP 协议使用统一的网络地址分配的方案，网络中的每台电脑都具有唯一的 IP 地址。

4. TCP/IP 协议是一个标准的高层协议，拥有极高的可靠性，可以为用户提供可靠的服务。

参 考 文 章 :  
[https://blog.csdn.net/my\\_learning\\_road/article/details/79941273](https://blog.csdn.net/my_learning_road/article/details/79941273)

## 5、说说 Aop 和 IOC 的应用

IOC 的主要应用场景体现在 BeanFactory 接口，BeanFactory 下面有具体的实现类来实现 IOC 的功能。

AOP 的主要应用场景：日志、权限、事物等。

## 6、Spring 中 bean 是线程安全的吗？

Spring 容器中的 Bean 本身不具备线程安全的特性，但是具体还是要结合具体 scope 的 Bean 去研究。

1. Spring 容器中的 Bean 默认是单例的，所有线程都共享一个单实例的 Bean，因此是存在资源的竞争。如果单例 Bean 是一个无状态 Bean，也就是线程中的操作不会对 Bean 的成员执行查询以外的操作，那么这个单例 Bean 是线程安全的。比如 Spring mvc 的 Controller、Service、Dao 等，这些 Bean 大多是无状态的，只关注于方法本身。对于有状态的 bean，是线程不安全的，但是我们可以通过 ThreadLocal 去解决线程安全的方法。

2. 对于原型 Bean（即 scope="prototype"），每次创建一个新对象，也就是线程之间并不存在 Bean 共享，自然是不会有线程安全的问题。

参考文章：[https://blog.csdn.net/qq\\_29645505/article/details/88432001](https://blog.csdn.net/qq_29645505/article/details/88432001)

## 7、Maven 中 package 和 install 区别

package 是把 jar 打到本项目的 target 下，而 install 时把 target 下的 jar 安装到本地仓库，供其他项目使用。

## 8、遇到过哪些设计模式？

参考文章：[https://www.cnblogs.com/Leo\\_wl/p/6067267.html](https://www.cnblogs.com/Leo_wl/p/6067267.html)

## 9、谈谈简单工厂和抽象工厂的区别

简单工厂模式：是由一个工厂对象创建产品实例，简单工厂模式的工厂类一般是使用静态方法，通过不同的参数的创建不同的对象实例 可以生产结构中的任意产品，不能增加新的产品

抽象工厂模式： 提供一个创建一系列相关或相互依赖对象的接口，而无需制定他们具体的类，生产多个系列产品 生产不同产品族的全部产品，不能新增产品，可以新增产品族

参考文章：<https://www.cnblogs.com/gclokok/p/10029088.html>

## 10、说说你常用的 Linux 基本操作命令

ls - 用来显示目标列表

cd - 用来切换工作目录

pwd - 以绝对路径的方式显示用户当前工作目录

cat - 文件内容查看

grep - 是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来

tail - 输出文件中的尾部内容  
ps - 用于报告当前系统的进程状态  
kill - 命令用来删除执行中的程序或工作  
top - 可以实时动态地查看系统的整体运行情况

## 11、谈谈 CurrentHashMap 的扩容机制

### JDK1.7:

先对数组的长度增加一倍，然后遍历原来的旧的 table 数组，把每一个数组元素也就是 Node 链表迁移到新的数组里面，最后迁移完毕之后，把新数组的引用直接替换旧的。

### JDK1.8:

扩容时候会判断这个值，如果超过阈值就要扩容，首先根据运算得到需要遍历的次数 i，然后利用 tabAt 方法获得 i 位置的元素 f，初始化一个 forwardNode 实例 fwd，如果 f == null，则在 table 中的 i 位置放入 fwd，否则采用头插法的方式把当前旧 table 数组的指定任务范围的数据给迁移到新的数组中，然后 给旧 table 原位置赋值 fwd。直到遍历过所有的节点以后就完成了复制工作，把 table 指向 nextTable，并更新 sizeCtl 为新数组大小的 0.75 倍，扩容完成。在此期间如果其他线程的有读写操作都会判断 head 节点是否为 forwardNode 节点，如果是就帮助扩容。

参考文章：<https://blog.csdn.net/axibaaa/article/details/97109455>