### 1.16 其它Java部分有关面试题

* 1.为什么局部内部类访问局部变量需要final?

[点击查看答案](https://www.cnblogs.com/oldpub-blog/p/9026824.html)

* 2.String、StringBuffer、StringBuilder、CharSequence的区别。

[点击查看答案](https://my.oschina.net/liululee/blog/1855342) [String和CharSequence区别](https://www.cnblogs.com/skywang12345/p/string01.html)

* 3.equals和==的区别？

[点击查看答案](https://www.cnblogs.com/smyhvae/p/3929585.html)

* 4.关于字符串的拼接你在项目中常常怎么操作的？为什么不能用“+”的方式进行拼接呢？

[点击查看答案](https://www.cnblogs.com/twzheng/p/5923642.html)

* 5.你觉得Lock和Synchronized的区别是什么？

[点击查看答案](https://blog.csdn.net/wu1226419614/article/details/73740899)

* 6.什么是Callback,讲讲你项目中使用的一些有关Callback的栗子。

这个题目就很简单了，回调在学Java基础就应该接触过，笔者就不提供答案了。

* 7.retrun & break & continue 区别？

[点击查看答案](https://blog.csdn.net/xiaomgee/article/details/78570139)

* 8.如何判断一个字符串是回文字符串？

[点击查看答案](https://www.cnblogs.com/dongtian/p/5665118.html)

* 9.什么是动态代理 & 什么是静态代理？

[点击查看答案](https://www.jianshu.com/p/b5e340ec9551)

* 10.String为什么会加final？

1 主要是为了“效率” 和 “安全性” 的缘故。若 String允许被继承, 由于它的高度被使用率, 可能会降低程序的性能，所以String被定义成final。 2、不允许其他类继承。这个应该不是最终原因，但这里权且也当成是一个原因。 3、 String类中的成员属性也几乎都设计成了private final的，这样String就被设计成一个不变类，这样有助于共享，提高性能。可以将字符串对象保存在字符串常量池中以供与字面值相同字符串对象共 享。如果String对象是可变的，那就不能这样共享，因为一旦对某一个String类型变量引用的对象值改变，将同时改变一起共享字符串对象的其他 String类型变量所引用的对象的值。 4、String被设计为不变类，其中的offset，value[]都被设计成private final的，这样在多线程时，对String对象的访问是可以保证安全的。java程序语言的许多特性依赖于不可变的String对象。

* 11.OOM可以try{}catch{}吗？
1. oom异常一般是java程序在做内存分配时，发现没有足够的剩余空间可用而抛出的异常；
2. 此时的分配空间可能是出于代码的new操作（用户主动），可能是出于内存的复制操作（语言自动），也可能是出于内存数据的重振操作（语言自动），可能是出jvm检测到外部信号（jvm自动）；
3. oom只是被建议为不要捕获的异常，因为通常你对这种情况是无能为力的！但你如果实在要捕获，why not ？
4. oom一般只会影响当前线程，而jvm中只要存在一个非daemon线程在运行，jvm就不会退出；
5. 如果是线程池运行环境，一般需要一个统一管理oom的程序，否则不能及时统一处理oom；

[分析链接](https://www.cnblogs.com/yougewe/p/9905754.html)

* 12.给我谈谈正则表达式。

[点击查看答案](http://www.cnblogs.com/zery/p/3438845.html)

* 13.如何将String转成int?

[点击查看答案](http://www.cnblogs.com/bluestorm/p/3698810.html)

* 14.谈谈你对String的理解。

这题和第10题有点联系,要从几方面入手，第一方面你可以谈它为什么设计为final的，第二方面为什么它是不可变的，你也可以从jvm角度分析它在内存中如何保存的都可以，笔者也就不提供答案了。

* 15.你如何理解序列化？有哪些方式序列化？

[点击查看答案](https://www.cnblogs.com/wxgblogs/p/5849951.html)

* 16.谈谈你对依赖注入的理解。

[点击查看答案](https://www.cnblogs.com/qiangweikang/p/4865486.html)

* 17.给我谈谈你对分派的理解。

[点击查看答案](https://www.cnblogs.com/hapjin/p/9374269.html)