Java初级面试题

1. synchronized 和 Lock 有什么区别?
   - 首先synchronized是Java内置关键字，在JVM层面，Lock是个Java类；
   - synchronized 可以给类、方法、代码块加锁；而 lock 只能给代码块加锁。
   - synchronized 不需要手动获取锁和释放锁，使用简单，发生异常会自动释放锁，不会造成死锁；而 lock 需要自己加锁和释放锁，如果使用不当没有 unLock()去释放锁就会造成死锁。
   - 通过 Lock 可以知道有没有成功获取锁，而 synchronized 却无法办到。
2. 说一下 HashMap 的实现原理?
3. == 和 equals 的区别?
   - ==：它是一种基本的数据比较操作符，用于比较两个对象的引用（内存地址）是否相等
   - equals()：它是 object 类中的方法，通常用于定义对象之间的逻辑相等性。
4. 八种基本数据类型有哪些？以及他们的封装类?
   - 八种基本数据类型：boolean、byte、short、int、float、double、long、char
   - 封装类分别是：Boolean、Byte、Short、Integer、Float、Double、Long、Character
5. Override 和 Overload 的含义和区别?
   - Overload 重载，方法名相同，参数列表不同（个数，顺序，类型不同），与返回类型无关；
   - Override 重写，将父类的方法覆盖。方法名相同，方法参数个数，顺序和类型都个父类相同；访问修饰符只能大于被重写的方法访问修饰符；
6. volatile 能保存原子性吗？为什么?
   - 不可以，volatile 关键字能够保证变量的可见性，但是不能保证原子性。
7. 接口与抽象类的区别?
   - 抽象类
     - 抽象类可以包含抽象方法和非抽象方法。抽象方法是没有实际实现的方法，需要子类去实现。非抽象方法有默认实现，子类可以选择性的重写方法；
     - 一个类只能继承一个抽象类
     - 抽象类可以包含成员变量，可以有构造函数，可以拥有普通方法；
     - 抽象类可以有访问控制修饰符，可以定义成员变量，可以包含构造方法
   - 接口
     - 接口中所有的方法都是抽象方法，没有方法体
     - 一个类可以实现多个接口
     - 接口中的成员变量隐式地是 static 和 final 的
     - 接口不能包含构造函数

2039字

- 接口中的方法默认发是 public 的，可以省略访问控制符，不能使用其他访问修饰符。

8. Java 创建对象有哪几种方式?

- new 创建新对象
- 通过反射机制
- 采用 clone 机制
- 通过序列化机制

9. Java 中有几种方法可以实现一个线程?

- 继承 Thread 类
- 实现 Runnable 接口
- 实现 Callable 接口，需要实现 call() 方法

10. 说一下 Runnable 和 Callable 有什么区别?

- 相同点:
  - 都是接口
  - 都采用 Thread.start() 启动线程
- 不同点:
  - Runnable 接口 run 方法无返回值；Callable 接口 call 方法有返回值，是一个泛型，和 Future、FutureTask 配合可以用来获取异步执行的结果
  - Runnable 接口 run 方法只能抛出运行时异常，且无法捕获处理；Callable 接口 call 方法允许抛出异常，可以获取异常信息

11. 线程状态有哪些，如何让线程中断?

a. 新建状态：当用 new 关键字创建一个线程对象后，该线程对象就处于新建状态。此时它已经有了相应的内存空间和其他资源，但是还没有开始执行。

b. 就绪状态：当线程对象调用了 start() 方法后，该线程就进入就绪状态。此时，线程已经准备好运行，但是是否真正执行取决于操作系统的调度。

c. 运行状态：当线程获得 CPU 资源并执行任务时，它就处于运行状态。

d. 阻塞状态：线程因为某种原因（如等待锁）放弃 CPU 使用权，暂时停止运行。当线程处于阻塞状态时，它不会占用任何 CPU 资源。

e. 等待状态：线程通过调用对象的 wait() 方法进入等待状态。此时，线程需要等待其他线程做出一些特定动作（如通知）。

f. 超时等待状态：线程通过调用对象的 sleep()、wait() 或线程的 join() 方法，或者是等待某个已触发的 Thread.sleep、Object.wait、Thread.join 等结构的超时时间到达，进入超时等待状态。

g. 终止状态：当线程因为 run() 方法执行完毕或者因为异常而退出，线程就处于终止状态。

要让线程中断，Java 提供了几种方法:

- 使用 interrupt() 方法，调用线程的 interrupt() 方法并不会直接停止线程，而是会在线程中设置一个中断状态。线程需要定期检查这个中断状态。
- 使用 stop() 方法：虽然 Java 提供了 stop() 方法，但这个方法已经被废弃，不建议使用。因为它会立即停止线程，这可能导致线程中的数据不一致，或者线程未完成的清理工作得不到完成（如文件、数据库等的关闭）
- 使用退出标志：另一种方法是使用退出标志。在线程的 run() 方法中，你可以定期检查一个共享变量（即退出标志），如果该变量表示线程应该退出，则线程可以通过正常的方法（如完成当前循环或任务）来结束执行。

12．ArrayList 和 LinkedList 有什么区别?

- ArrayList是基于索引的数据接口，它的底层是数组。它可以以O(1)时间复杂度对元素进行随机访问。与此对应，LinkedList是以元素列表的形式存储它的数据，每一个元素都和它的前一个和后一个元素链接在一起，在这种情况下，查找某个元素的时间复杂度是O(n)。
- 相对于ArrayList，LinkedList的插入，添加，删除操作速度更快，因为当元素被添加到集合任意位置的时候，不需要像数组那样重新计算大小或者是更新索引。
- LinkedList比ArrayList更占内存，因为LinkedList为每一个节点存储了两个引用，一个指向前一个元素，一个指向下一个元素

13．说下 JVM 的内存区域?

14．垃圾收集算法有哪些?

- 标记-清除算法：标记”和“清除”两个阶段；首先标记出所有需要回收的对象，在标记完成后统一回收掉所有的标记对象。
- 复制算法：将可用内存按容量划分为大小相等的两块，每次只使用其中的一块。当一块的内存用完了，就将还存活的对象复制到另一块上面，然后再把已使用过的内存空间清理掉。
- 标记-压缩算法：标记过程跟“标记-清除”算法一样，但后续步骤不是直接对可回收对象进行清理，而是让所有存活的对象都向一端移动，然后清理掉端边以外的内存。

15．