Java面试-字符串:

`String`、`StringBuffer` 和 `StringBuilder` 的区别

- **String类是不可变类**，任何对`String`的改变都会引发新的`String`对象的生成；
- `StringBuilder`是**非同步**,运行于多线程中就需要使用着单独同步处理。
- 字符串之间的频繁操作的话就用`StringBuffer`，`StringBuffer`支持并发操作，**线性安全**的，适合多线程中使用

扩展：`StringBuffer`为什么线程安全？因为`StringBuffer`很多方法都是`synchronized`修饰的

`String` 常用方法

- `charAt(int index)`：获取指定索引位置的字符
- `indexOf(int ch)`：返回指定字符在此字符串中第一次出现处的索引。
- `indexOf(String str)`：返回指定字符串在此字符串中第一次出现处的索引。
- `indexOf(int ch,int fromIndex)`:返回指定字符在此字符串中从指定位置后第一次出现处的索引。
- `substring(int start,int end)`：从指定位置开始到指定位置结束截取字符串。
- `equals(Object obj)`：比较字符串的内容是否相同,区分大小写
- `equalsIgnoreCase(String str)`：比较字符串的内容是否相同,忽略大小写
- `isEmpty()`：判断字符串的内容是否为空串""。
- `getBytes()`：把字符串转换为字节数组。
- `toCharArray()`：把字符串转换为字符数组。
- `valueOf(char[] chs)`：把字符数组转成字符串。
- `valueOf(int i):` 把int类型的数据转成字符串。（String类的valueOf方法可以把任

意类型的数据转成字符串。）

- `toLowerCase()`: 把字符串转成小写。
- `toUpperCase()`: 把字符串转成大写。
- `concat(String str)`: 把字符串拼接。
- `replace(String old,String new)`: 将指定字符串进行互换

`Object` 常用方法

- `clone()`: 创建并返回此对象的一个副本。需要实现 `Cloneable` 接口
- `equals(Object obj)`: 指示某个其他对象是否与此对象“相等”。
- `finalize()`: 当垃圾回收器确定不存在对该对象的更多引用时，由对象的垃圾回收器调用此方法。
- `getClass()`: 返回一个对象的运行时类。
- `int hashCode()`: 返回该对象的哈希码值。
- `notify()`: 唤醒在此对象监视器上等待的单个线程。
- `notifyAll()`: 唤醒在此对象监视器上等待的所有线程。
- `toString()`: 返回该对象的字符串表示。
- `wait()`: 导致当前的线程等待，直到其他线程调用此对象的 notify() 方法或 notifyAll() 方法。
- `wait(long timeout)`: 导致当前的线程等待，直到其他线程调用此对象的 notify() 方法或 notifyAll() 方法，或者超过指定的时间量。
- `void wait(long timeout, int nanos)`：`nanos` 表示额外的等待时间，单位是纳秒，当 `nanos` 大于0时，`timeout` 会自动加1，以保证至少多等1毫秒

`==` 和 `equals` 的区别是什么？

`==` 对于基本类型和引用类型的作用效果是不同的

- 基本类型：比较的是值是否相同；
- 引用类型：比较的是引用是否相同；

`equals` 默认情况下是引用比较，只是很多类**重写**了 `equals` 方法，比如 `String`、`Integer` 等把它变成了值比较，所以一般情况下 `equals` 比较的是值是否相等。