引用类型究竟时如何实现数据的保存与操作?

---栈内存（stack）：存放对象实体的地址（相当于c中的指针，但是注意指针变量和指针变量的值，所以变量名和变量值一一对应++++++键值对）

---堆内存(heap)：存放的是每一个对象的具体内容（即是对象实体，***属性-filed及其值***）（new的作用就是在堆中开辟一个内存空间并返回一个地址返回给引用）

**（Java中对象内存开辟有很多种方法，但是new是Java底层给予最直接最简单的内存方法）**

**栈引用堆**

**对象实例化的方式两种，分步和不分步，前者需要经历一个指向null的步骤，当真正new时一致。**

**只有引用会出现“NULLPOINTEREXCEPTION（空指针异常）”**

**同一个栈内存可以同时指向同一块堆内存。**

**没有任何引用指向的堆内存，就会成为垃圾，垃圾会被jvm的垃圾回收器定期进行清理，但是，频繁凡人垃圾回收会引起严重的性能下降，所以要避免无用内存的开辟。**

**方法不在堆内存中，方法在方法区中，再细点就是方法区的class对象中。**

数组和对象本身在堆中分配，即使程序运行到使用 new 产生数组或者对象的语句所在的代码块之外，数组和对象本身占据的内存不会被释放，数组和对象在没有引用变量指向它的时候，才变为垃圾，不能在被使用，但仍然占据内存空间不放，在随后的一个不确定的时间被垃圾回收器收走（释放掉）。这也是 Java 比较占内存的原因

**java中常用的内存区域**

在java中主要存在4块内存空间，这些内存的名称及作用如下：

1. 栈内存空间：保存所有的对象名称（更准确地说是保存了引用的堆内存空间的地址）
2. 堆内存空间：保存每个对象的具体属性内容。
3. 全局数据区：保存static类型的属性。
4. 全局代码区：保存所有的方法定义。