

# Java值传递的一些理解

今天下午一直在看一些常见的Java试题，看到很多问题都涉及到Java参数传递的问题，之前自己也是一知半解，虽然知道是值传递，但没有办法通俗易懂的讲出个所以然来，所以特地查一些资料并且记录下来今天的理解。首先，Java的内存分配分为三种：栈内存、堆内存、方法区内存。

- 1.栈内存：主要存放函数中定义的基本类型变量，对象的引用。在程序运行到跳出变量作用域后Java会自动释放内存。使用栈的好处是在于，栈的先进后出的性质符合程序中函数调用运行特点，即后调用先返回，这样就可以尽快释放不用到的内存空间。而且栈内存一般使用的是 一级cache，读写速度快。
- 2.堆内存：主要存放由new 创建的对象和数组，Java垃圾回收器回收这部分内容。使用堆结构的意义，我推测是在于易于控制（垃圾回收的时候总是回收最久没有使用的空间，而堆可以很快的取出结构中最大或最小的值）。
- 3.方法区内存：主要存放类信息和static变量。这里穿插一个小知识点：在每次类被加载的时候，总是先执行static属性的定义，以及static语句的执行。

好，说完内存就可以清晰的解释Java参数传递了。

- 1.Java只用值传递
- 2.值传递时只复制了栈内存的内容
- 3.对于基本类型对象， 栈内存中留有的是原值，所以参数传递时，值传递将原值复制一份到调用函数内
- 4.对于复杂对象（ new 出来的对象，通过单例模式获得的对象我认为也算是new 出来的对象，或者说是原值在堆内存中的对象），栈内只存有的是一个指向堆内值得引用。所以虽然值传递复制的是栈内的“原值”，但表现出来的特性仍然是趋向于引用传递（因为你复制一个引用的原值，并没有卵用）。

总之，传参数时，如果参数是基本类型，那在调用之后原值不会被改变；如果参数是复杂类型，在调用后参数会改变。

参考资料是知乎的一个问题下的众多答案：

<https://www.zhihu.com/question/19569932>