C++的指针 和 Java中的引用 &C++的指针

指针就是指针变量 --- 指针变量重新赋值 ---- 就表示这个指针从一个变量指向了另外一个变量 ----- 因为指针本身的值不同 表示地址已经变化了 ---- 指向的变量变化了

引用本质是常指针 ---- 就是只能指向同一块区域的指针 ---- 用法升级了 不用指针符号 使用简单 ----- 也就是引用一旦初始化 引用到某一个变量上面 就不能再改变引用本身的内容 来指向其他变量了 ---- 引用的变量本身可以修改

指针 和 引用 具体是指向或者引用栈内存 还是堆内存 还是常量区 的变量 还是对象 都有可能

但是 java ----- 根据使用的java的习惯

A a = new A();

a在java中叫做引用 ------ 那么 a会不会被改变呢？

a=new A(1,2); ----同样编译没有问题 ----- 说明 a是可以修改本身内容的java引用 ----- 但是C++中的引用本身内容不可以修改 ------ 所以 java中的引用从本质上更像C++中的指针 ----- C++中指针变量可以修改 同样 java中引用可以修改

----- 声明上：但是 Java中的引用的好处就是 不用A\* 这样的标记 去掉了\* 也没有使用&表示引用 就是直接是类型A ----- 声明上比C++中的引用和指针都方便

------ 使用上 传递java的引用 可以直接修改java引用指向的对象 ----- C++传递指针 和 传递引用都可以 但是 C++传递引用在调用时候的可读性和执行方法内部的使用上都是非常方便的 但是 C++传递指针没有那么方便

----- Java的引用是继承了C++中引用的使用上的方便性 去除了C++中引用的限制 和 C++中引用定义的麻烦

**=== 综上所述 Java的引用是继承了C++中 指针和引用各自的优点的产物 ---- 本质更像C++指针 使用上更像C++引用**

<http://www.cnblogs.com/Mr-xu/archive/2012/08/07/2626973.html>