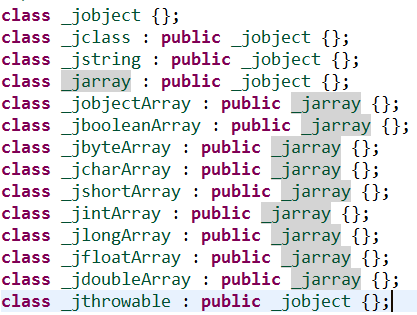
1. #define JNI\_FALSE   0
2. #define JNI\_TRUE    1
4. #define JNI\_VERSION\_1\_1 0x00010001
5. #define JNI\_VERSION\_1\_2 0x00010002
6. #define JNI\_VERSION\_1\_4 0x00010004
7. #define JNI\_VERSION\_1\_6 0x00010006
9. #define JNI\_OK          (0)         /\* no error \*/
10. #define JNI\_ERR         (-1)        /\* generic error \*/
11. #define JNI\_EDETACHED   (-2)        /\* thread detached from the VM \*/
12. #define JNI\_EVERSION    (-3)        /\* JNI version error \*/
14. #define JNI\_COMMIT      1           /\* copy content, do not free buffer \*/
15. #define JNI\_ABORT       2           /\* free buffer w/o copying back \*/

Java中有两类数据类型：primitive types，如，int, float, char；另一种为reference types，如，类，实例，数组。

 注意：数组，不管是对象数组还是基本类型数组，都作为reference types存在，有专门的JNI方法取数组中每个元素。



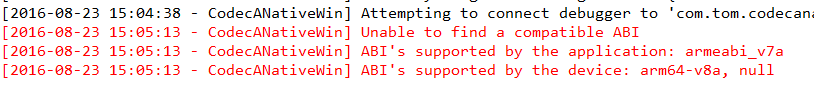
1.这些通过JNI传递到Native层 函数参数 ，都是LocalReference 要使用env->NewGlobalRef

2.jFieldID和jMethodID都不用 以为他们只是一个struct 不是jobject





各种类型的使用 [http://blog.csdn.net/conowen/article/details/7523145](http://blog.csdn.net/conowen/article/details/7523145%20)



[ro.product.cpu.abi]: [arm64-v8a]

[ro.product.cpu.abilist32]: [armeabi-v7a,armeabi]

[ro.product.cpu.abilist64]: [arm64-v8a]

[ro.product.cpu.abilist]: [arm64-v8a,armeabi-v7a,armeabi]