

MICROOH 麦可网

Android-从程序员到架构师之路

出品人：Sundy

讲师：高焕堂（台湾）

<http://www.microoh.com>

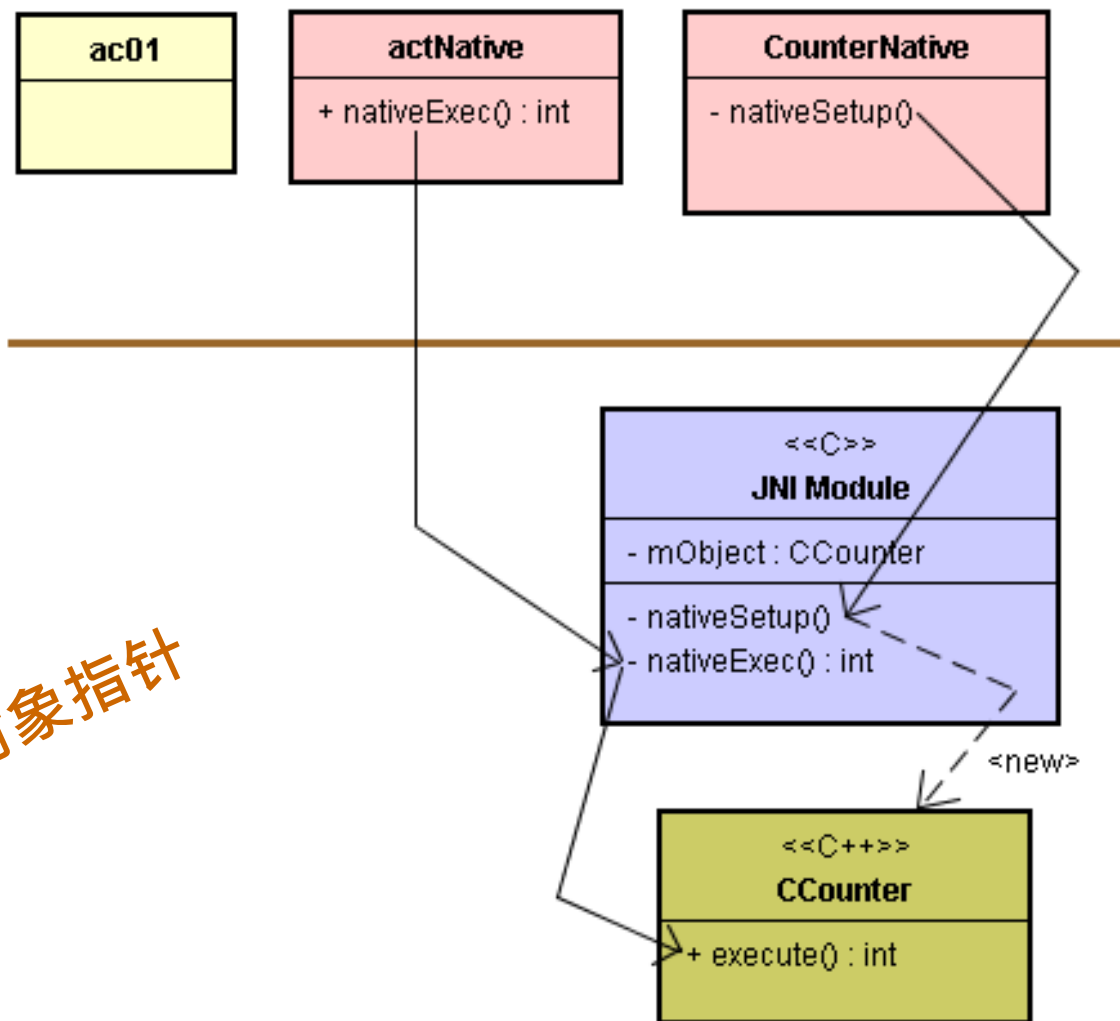
C04_b

JNI：必要的优化设计(b)

By 高煥堂

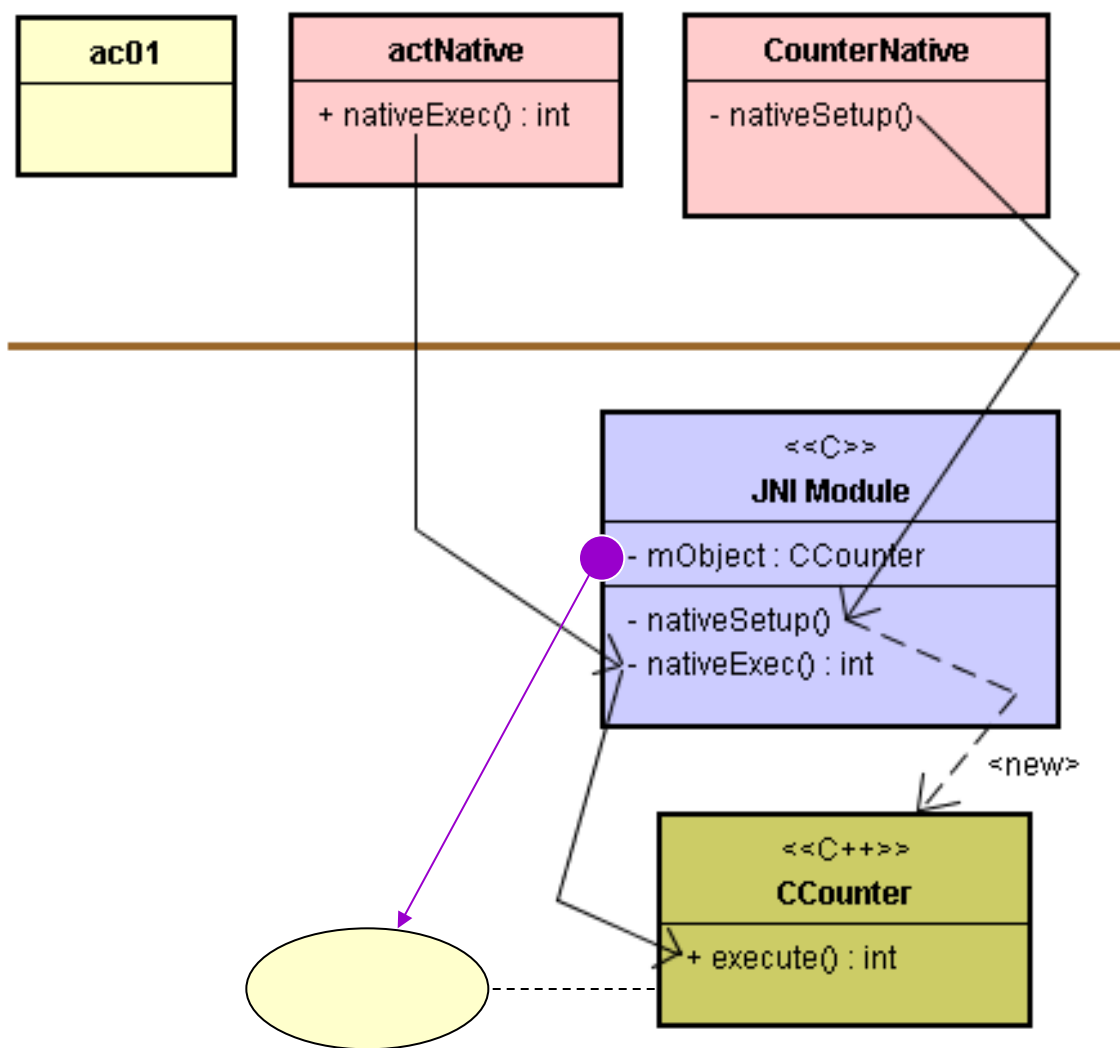
3、<静态对静态，
动态对动态>原则

- 在JNI的C模块里，不仅能创建Java层的对象，也可以创建C++类的对象。
- 但是，要将CCounter类新对象的指针放在那里才合理呢？



CCounter对象指针
放那里?

- 这nativeSetup()函数动态创建CCounter类的对象，并将新对象的指针储存于全局(静态)的mObject变量里。



- 接着，nativeExec()函数透过mObject变量的指针值而调用CCounter的execute()函数。


```
/* com.misoo.counter.CounterNative.cpp */
#include "com_misoo_counter_actNative.h"
#include "com_misoo_counter_CounterNative.h "

class CCounter {
    int n;
public:
    CCounter(int v) { n = v; }
    int execute()
    { int i, sum = 0;
      for(i=0; i<=n; i++) sum+=i;
      return sum; }
} *mObject;
```

```
JNIEXPORT void JNICALL  
Java_com_misoo_counter_CounterNative_nativeSetup  
    (JNIEnv *env, jobject thiz) {  
    mObject = new CCounter(10);  
}
```

```
JNIEXPORT jint JNICALL  
Java_com_misoo_counter_actNative_nativeExec  
    (JNIEnv *env, jclass clazz) {  
    return (jint) mObject->execute();  
}
```

违背

静态对静态，
动态对动态

原则



~ Continued ~