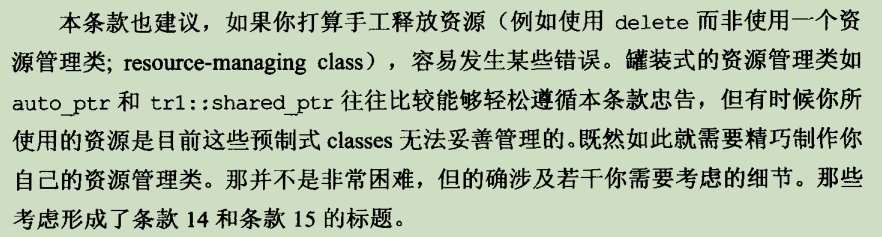


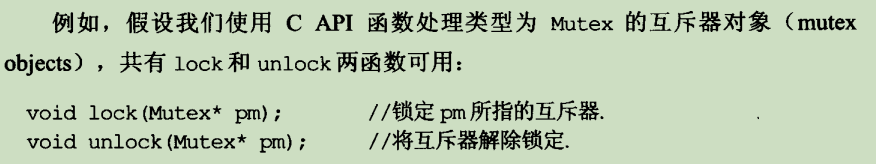
**条款13**

auto\_ptr 和shared\_ptr

auto\_ptr (不能够复制)

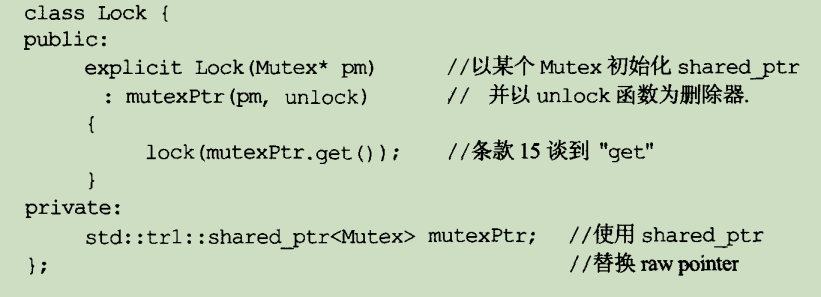


**条款14**



如果我们想要对其进行复制操作，有两种想法，第一我们不希望他复制（这是一种绑定操作），第二种如果复制（这些复制的lock都是绑定到同一个对象上），我希望把这些lock都解锁unlock，我们采用shared\_ptr的思想（计数器），但是shared\_ptr调用析构函数，我们想要的是unlock，所以我们书写一个对象管理

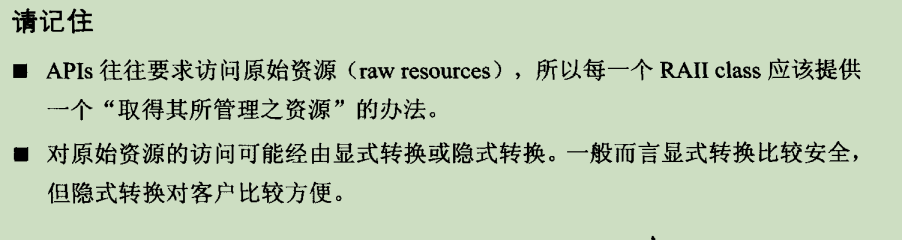
**解决方法**



把mutex申明成shared\_ptr，shared\_ptr在计数器为0的时候可以调用我们赋予他的所谓析构函数，

如果我们想要lock只能绑定一个，可以采用auto\_ptr的思路，与shared\_ptr一样操作

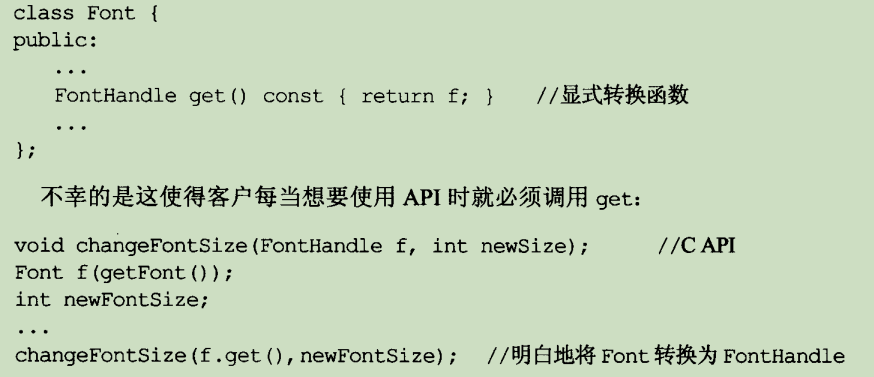
**条款15**



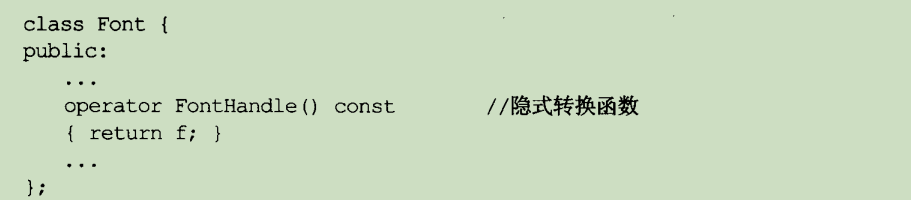
shred\_ptr 提供了一个get函数用于获取原始资源，如果某个class直接使用原始内存我们不推荐使用，对于shred\_ptr这种需求我们用两种获取方式

**1.显示转换**

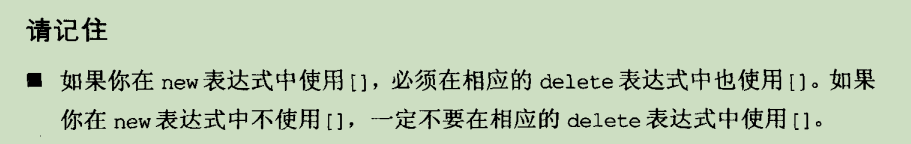
用一个get函数进行封装



**2.隐式转换（有时候会产生歧义，看书P72）**



**条款16**



**条款17**

