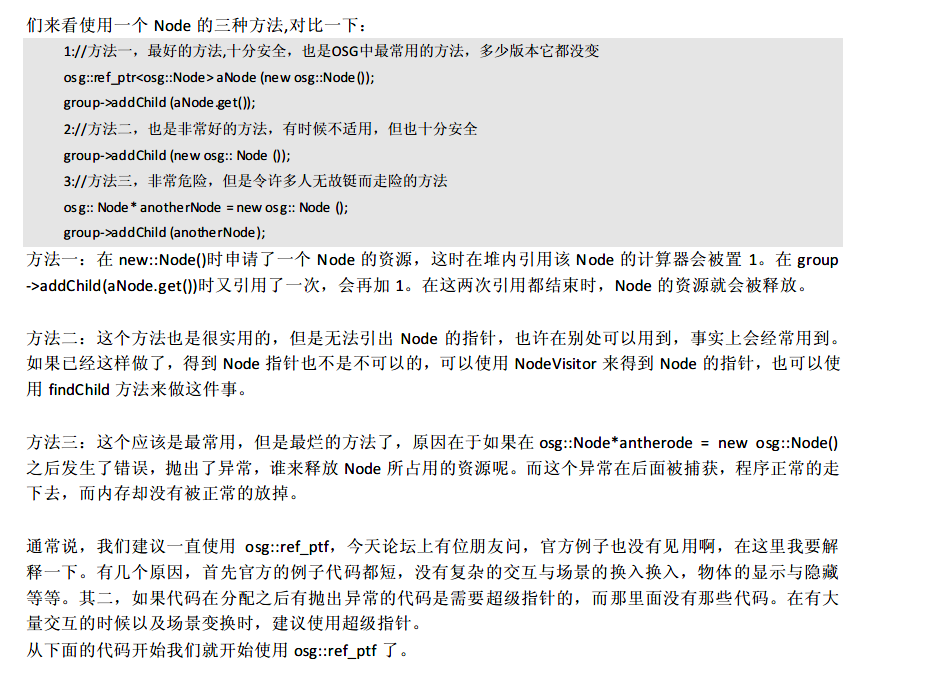
1. 有一点比较重要然后容易忽略的是，在osg程序开发中，请尽量使用osg::ref\_ptr<T\*>；这个智能指针构建起osg的对象树（绝大部分osg类都是继承于osg::Referenced，即可计数对象），类似于Qt的对象树，负责其内存的管理。当然其内存管理策略不是我们关心的，我们使用它的原因是基于其内存管理策略，大部分类的析构函数都是受保护的，意味着我们并不能直接维持对象的内存，就必须用到osg::ref\_ptr（当然这个必须是相对的，举个例子，如果我们new了一个osg::Node，然后把它addChild()进mapNode，那当removeChild的时候，这块内存其实也是会被释放掉的。但这责任分配很奇怪不是吗？）根据部分教材的说法：



这种时候就必须由我们来负责内存释放，而这是必须通过osg::ref\_ptr完成。

至于我们怎么释放？很简单，举例：osg::ref\_ptr<osg::Node> myNode;

直接myNode = nullptr;这块内存就会被释放（基于计数原则），同时把myNode reset为空。

Osg::ref\_ptr本身非常简单，除了上面所说基本没有其他内容（而正是基于这么简单的内容就构建起整个内存管理系统）。