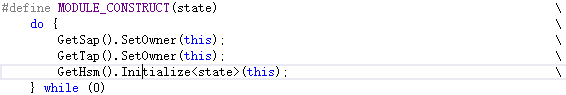
一般每个协议构造函数都执行这一句话：（在传入模板参数的时候，这个类就构造完成了，包括类内的Sap、Tap等模板类，而Sap、Tap中的owner\_指针是一个野指针，因此需要SetOwner进行初始化，即传入当前协议对象的指针this。）



module.h中的宏定义：



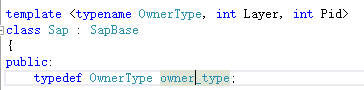
所有协议都继承module类，在**module类中**：





**当向Sap类传入模板的时候，就生成了模板类sap\_type，协议类通过模板类sap\_type来构造对象sap\_**。

Sap中对传入模板的利用：



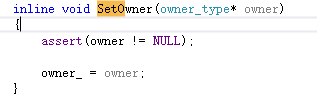




在**Sap类中**，构造函数如下，其中SapMap是单例的，并且Sap在构造函数就完成了注册（相当于绑定了当前Sap和SapMap的关系）



设置**Sap中**owner的函数（owner指的都是协议），



总结：每个协议都继承自module基类，module基类中有一个Sap对象，会把当前对象的类作为模板传入Sap的模板。而Sap对象构造的时候，就会调用Register函数与SapMap完成绑定，并且在module调用函数把Sap对象的owner设置成当前协议。（SapMap的映射是层ID和协议ID到一个SapBase指针的映射，因此调用Register函数时，实际上是把子类指针赋给父类指针。）

应用时：从SapMap映射中找到，然后调用Sap接口中的函数。

