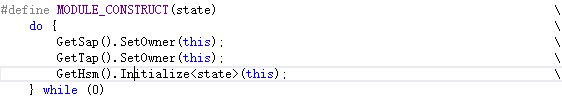
1. 一般协议的构造函数先执行这一句宏，对Sap、Tap和状态机进行绑定（设置它们的owner。）

2. 向协议类模板module传入模板时，就创建了模板类，包括Sap模板类。

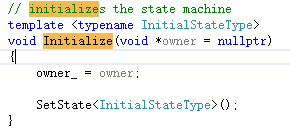
3. 构造完之后应该只是设置好了Top状态和初始状态。



在module.h中的宏：



**状态机类中**的初始化：



**状态机类中**的SetState函数：（这个函数可以跳一级状态，首先判断当前状态是否传入状态的外部状态，如果不是的话调用on\_exit。如果是，则说明是要进入状态，因此调用on\_enter）

1. GetState，把当前状态插入状态机的state\_list\_中。

2. 由于未初始化的current\_state\_是NULL，因此不执行。

3. 把当前状态机的状态设置为传入的状态。

4. 把传入的状态的is\_enter\_设置为true。

5. 调用该状态的on\_enter函数，因为相当于进入了这个状态。

6. 默认init参数是true，因此调用HasInitializeState()，如果未设置成初始状态，则递归调用SetState，把状态设置成协议中设定好的初始状态。

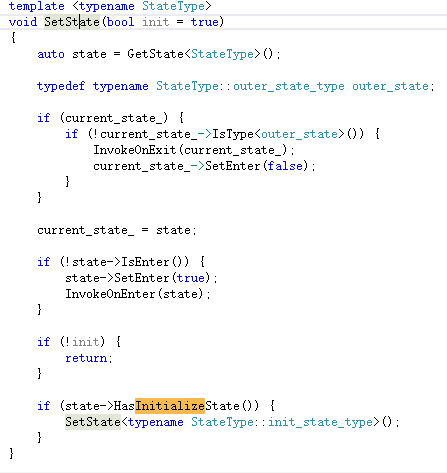
注意：

1. 如果是离开状态，则需要把原来的is\_enter\_设置为false。如果是进入状态，需要把进入之后的状态的is\_enter\_设置为true。

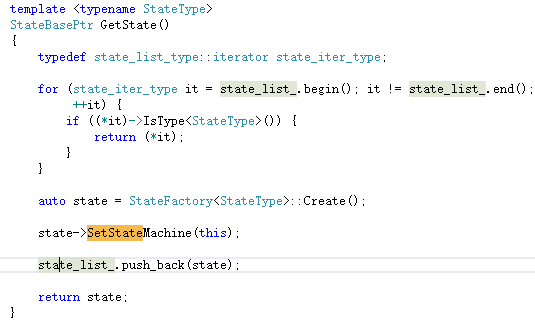
2. 每个状态的初始is\_enter\_都是false，表示没有进入该状态。（如果转移以后还是在父状态下，父状态仍为true。）

3. 每个状态机都有一个状态列表state\_list\_，因此每个状态的对象是唯一的。

4. 每个状态基类都有on\_enter函数，需要协议自己写。



GetState的作用是从state\_list\_中找这个状态，如果找不到这个状态就构造一个这个状态的指针，并且压入state\_list\_和返回。



查看当前状态是不是已经设置成初始状态。

