

```
//二：成员函数末尾的const
//const：常量，在成员函数屁股后边增加一个const。不但要在成员函数声明中增加const，也要在成员函数定义中增加const；
//作用：告诉系统，这个成员函数不会修改该对象里任何成员变量的值等等。也就是说，这个成员函数不会修改类Time的任何状态。
//屁股后边加一个const后缀的成员函数“常量成员函数”；

const Time abc: //定义const对象。这种对象有限制。
Time def:
//abc.initMillTime(1500): //不可以，initMillTime()并不是const成员函数，这个initMillTime()只能被非const对象调用。
//def.initMillTime(1500):
abc.noone(): //不可以调用noone()，因为noone()没有声明成const。
def.noone():

//const成员函数 则不管是const对象，还是非const，都可以调用const成员函数。
//而非const成员函数，不能够被const对象调用，只能被非const对象调用。
```

```
void Time::addhour(int tmphour) const
{
    //Minute = tmphour;
}
```

普通函数不是成员函数，也就不存在改变成员变量的值，所以普通函数不能使用后置的const

注意：如果由const后置在成员函数后面，那么成员函数的声明和定义都需要加后置const