Test Moq & Rhino Mocks

# Automate Test

<http://www.pluralsight.com/courses/automated-testing-fraidy-cats>

## Stub—设置Mock对象的属性值

# Common Problem with Unit Test

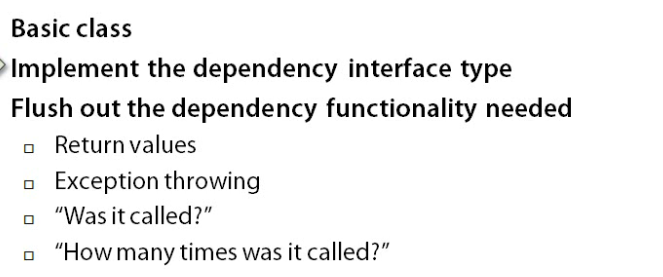
http://www.pluralsight.com/courses/rhinomock-fundamentals

测试对象依赖于一大堆其他对象，导致没法单元测试或者单元太复杂测试效果不好。

方法是用Fake对象取代测试对象依赖的一大堆对象

# 手动Mock

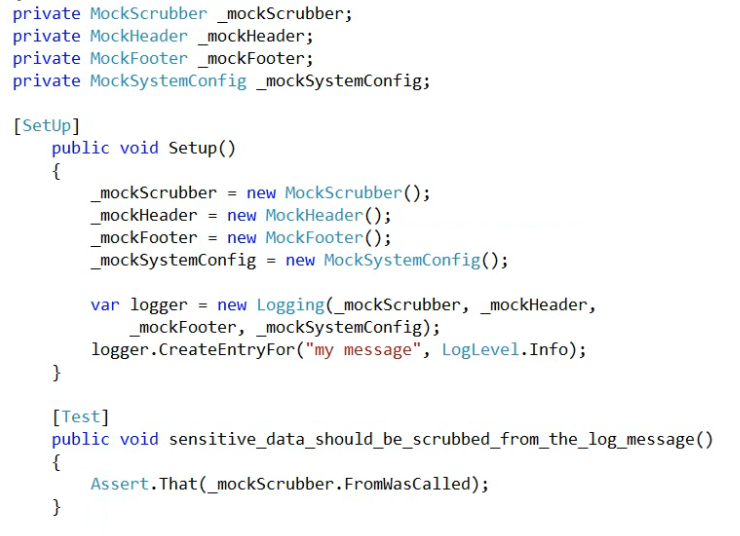
## 流程



简单点说就是在单元测试类的setup中初始化所有Mock，并把测试对象装好作为成员变量，在Test里用。

FromWasCalled可以检验测试对象是不是用到了应该用到的组件。

缺点是为了一小段代码要创建一大堆的Mock



# Rihno Mock

## GenerateMock

RihnoMock会根据interface自动生成一个Mock出来。

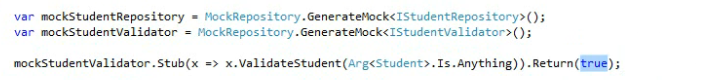
RihnoMock还会扩展没一个对象，加上一堆测试函数，比如AssertWasCalled.



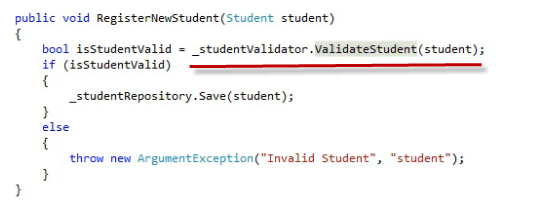
## Property Called



## Stub to control flow



这样就能测下面这个分支了



## Stub to 设置属性值

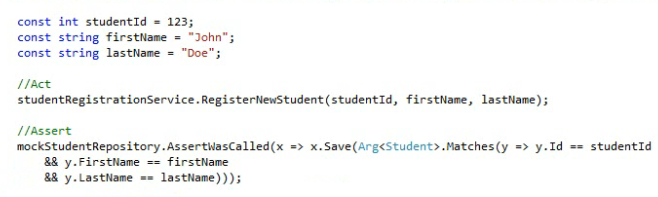


## Assert与Playback

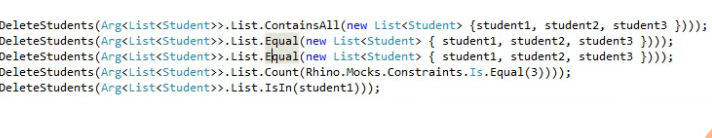


## Constraints—Matches

限制Assert的条件



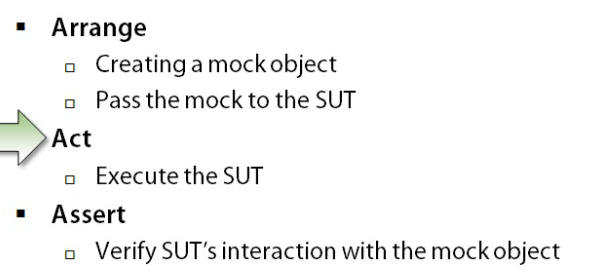
## Constraints—List



## AutoMocker—StructureMap

StructureMap是个IoC Container，和Rhino Mocks没关系，从NuGet拿

# Moq



## Demo

Mock一个ICustomerRepository，然后设置Save函数，再执行

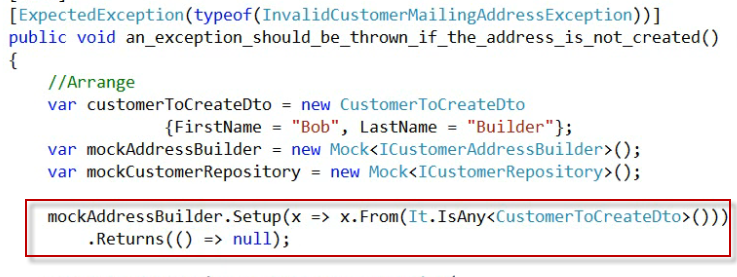


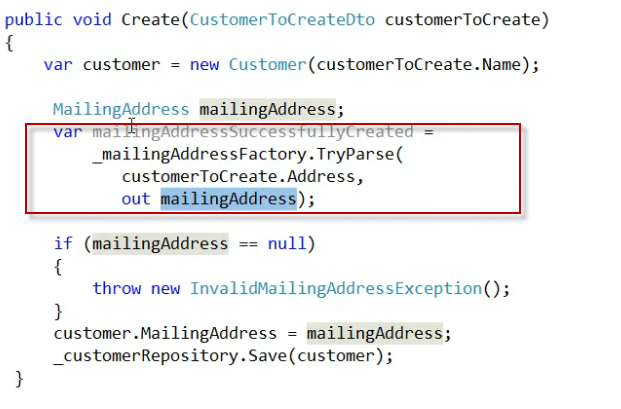


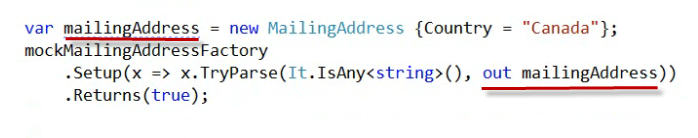
## Mock Return Values

测试目标代码是这样的，要Mock这个From函数，返回的是一个Address

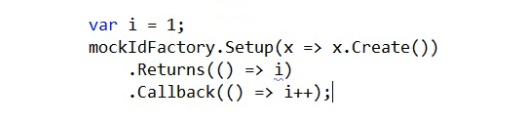






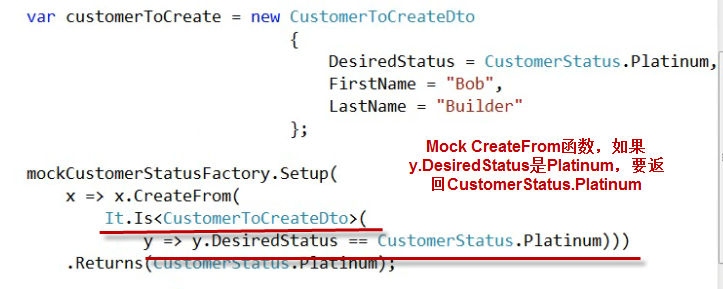


多次设置返回值，每次返回i++



## Arguments

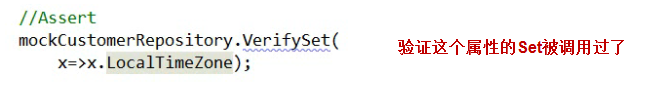
上面那个对象将被传到函数里，函数根据他的属性返回值



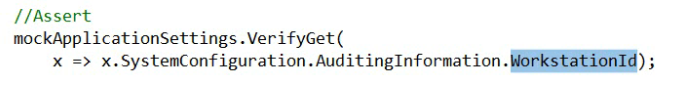
## Exceptions



## Property







## Stub—设置Mock对象的属性值

