

Моск-объекты

Практика

Юрий Литвинов
y.litvinov@spbu.ru

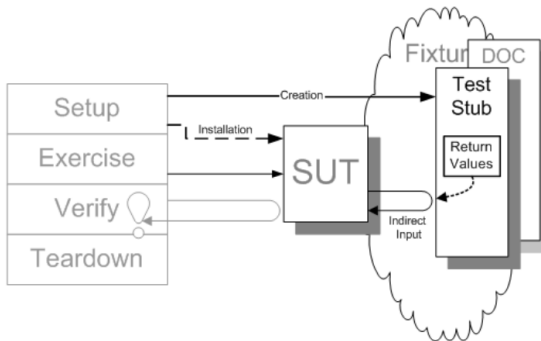
19.04.2024

Mock-объекты

- ▶ Объекты-заглушки, симулирующие поведение реальных объектов и контролирующие обращения к своим методам
 - ▶ Как правило, такие объекты создаются с помощью библиотек
- ▶ Используются, когда реальные объекты использовать
 - ▶ Слишком долго
 - ▶ Слишком опасно
 - ▶ Слишком трудно
 - ▶ Для добавления детерминизма в тестовый сценарий
 - ▶ Пока реального объекта ещё нет
 - ▶ Для изоляции тестируемого объекта
- ▶ Для mock-объекта требуется, чтобы был интерфейс, который он мог бы реализовать, и какой-то механизм внедрения объекта
 - ▶ Как правило, как параметр конструктора или через свойство

Заглушки (Stubs)

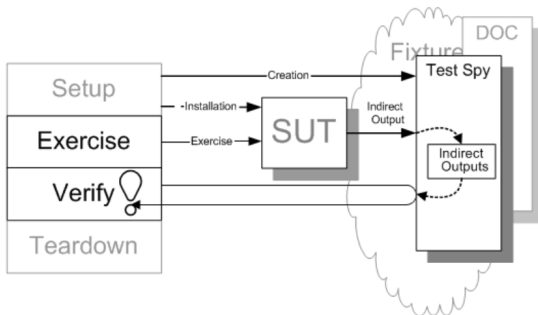
- ▶ Захардкоженные объекты, реализующие нужный интерфейс
- ▶ Наивная или вообще отсутствующая реализация
 - ▶ Возможно какие-то assert'ы или полезная логика для теста



© <http://xunitpatterns.com>

Spies

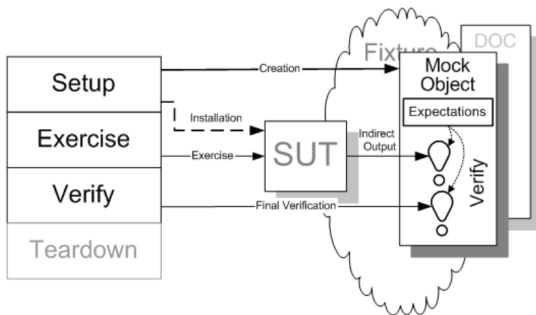
- ▶ Запуск теста, затем проверка условий и ограничений



© <http://xunitpatterns.com>

Mocks

► Конфигурирование объекта перед запуском теста



© <http://xunitpatterns.com>

Библиотеки

- ▶ Moq (<https://github.com/devlooped/moq>) — стандарт де-факто для моков в тестах на .NET
- ▶ NSubstitute (<https://nsubstitute.github.io/>) — несколько более «чистый» синтаксис
- ▶ FakeItEasy (<https://fakeiteasy.github.io/>) — достойная альтернатива, с местами более аккуратным, а местами более громоздким синтаксисом

Пример

Moq

/// Arrange

```
var mock = new Mock<ILoveThisLibrary>();  
mock.Setup(library => library.DownloadExists("2.0.0.0"))  
    .Returns(true);  
ILoveThisLibrary lovable = mock.Object;
```

/// Act

```
bool download = lovable.DownloadExists("2.0.0.0");
```

// Assert

```
mock.Verify(library  
    => library.DownloadExists("2.0.0.0"), Times.AtMostOnce());
```

© <https://github.com/devlooped/moq> (слегка адаптировано)

Более жизненный пример (1)

Moq

```
public class System(IDependency dependency)
{
    public int DoSomething(int x)
        => dependency.ImportantMethod(x);
}
```

```
public interface IDependency
{
    int ImportantMethod(int importantParameter);
}
```

```
public class Dependency: IDependency
{
    public int ImportantMethod(int importantParameter)
        => importantParameter + 1;
}
```


Более жизненный пример (2)

Moq

```
[Test]
public void TestWithMocks()
{
    var dependencyMock = new Mock<IDependency>();
    dependencyMock.Setup(dependency => dependency.ImportantMethod(1))
        .Returns(2);

    var system = new System(dependencyMock.Object);

    var result = system.DoSomething(1);

    Assert.That(result, Is.EqualTo(2));
    dependencyMock.Verify(dependency => dependency.ImportantMethod(1),
        Times.Exactly(1));
}
```

Задача

- ▶ Вспомнить домашнее задание про стековый калькулятор
- ▶ Переделать стеки так, чтобы они были генериками
- ▶ Написать тесты для калькулятора, изолированные от реализации стеков