Java JUnit
for Unit Testing

MOCKITO FRAMEWORK API FOR MOCKING

ERIC Y. CHOU, PH.D.   IEEE SENIOR MEMBER

Mockito

mockito.org

Target and challenge of unit testing

A unit test should test a class in isolation. Side effects from other classes or the system should be eliminated if possible. The achievement of this desired goal is typical complicated by the fact that Java classes usually depend on other classes.

To solve this, you can use test doubles. (Test Doubles is a Design Pattern, a known solution pattern.)

Classifications of different test classes

A dummy object is passed around but never used, i.e., its methods are never called. Such an object can for example be used to fill the parameter list of a method.

Fake objects have working implementations, but are usually simplified, for example they use an in memory database and not a real database.

A stub class is an partial implementation for an interface or class with the purpose of using an instance of this stub class during testing. Stubs usually do responding at all to anything outside what's programmed in for the test. Stubs may also record information about calls.

A mock object is a dummy implementation for an interface or a class in which you define the output of certain method calls.
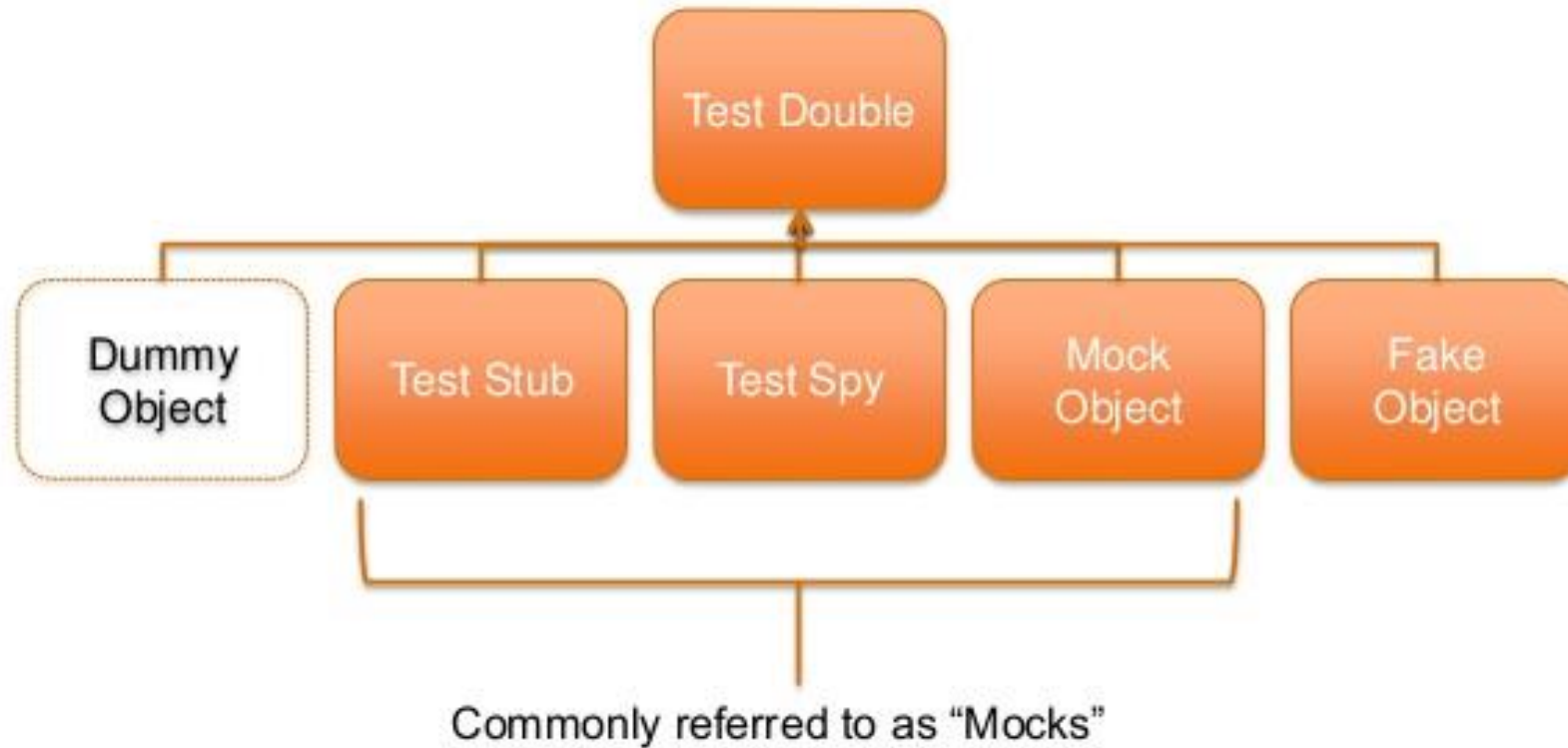
Background
Stubs, Mocks, Spy, Dummy and Fakes are Collaborators.

The first thing we need is a little background - when unit testing, which is primarily what people are talking about when discussing TDD, the class you are testing is called the **System Under Test (SUT)**. There are very few classes that operate entirely in isolation. Usually they need other classes or objects in order to function, whether injected via the constructor or passed in as method parameters. These are known as **Collaborators**. When we talk about Collaborators in the context of unit testing, we are specifically talking about Collaborators for the System Under Test.
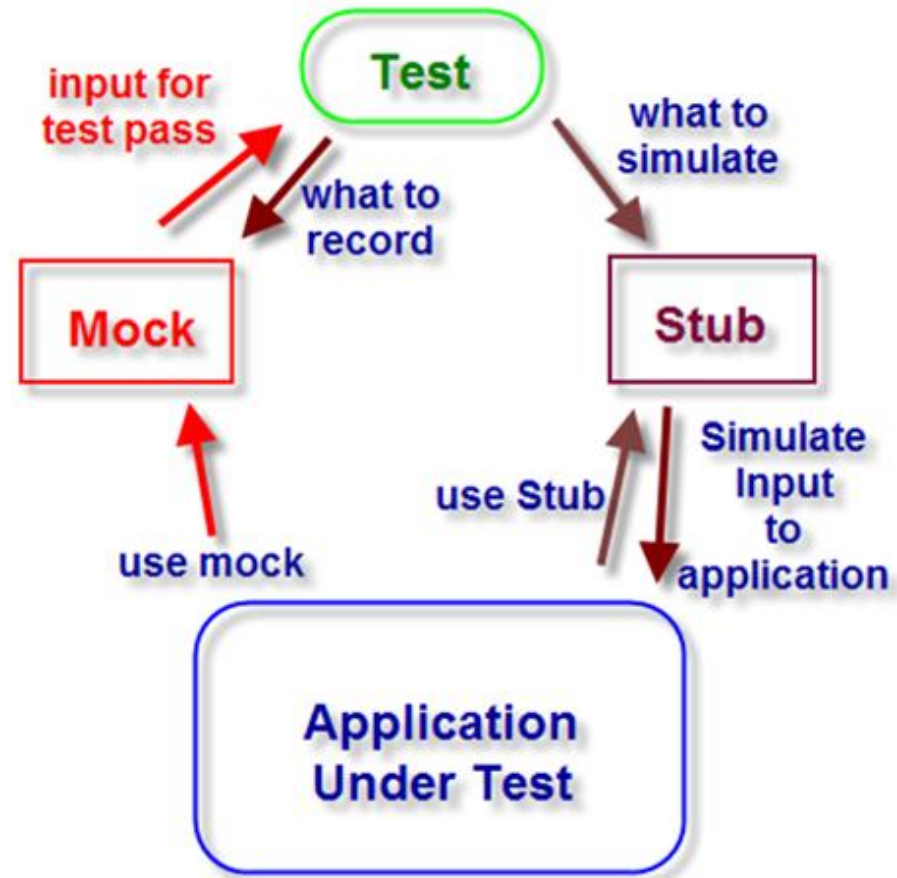
What Are Mocks?

Use substitutes for the tester (**Stubs**) or the SUT (**Mocks**) to perform **Unit Test**.

Stubs and Mocks

Mockito

Mockito is a test spy framework and it is very simple to learn. Notable with **Mockito** is that expectations of any mock objects are not defined before the test as they sometimes are in other mocking frameworks. This leads to a more natural style(IMHO) when beginning mocking.

Installation of mockito package

Go Chrome!!!

Go Eclipse!!!