## PowerMock - <https://code.google.com/p/powermock/>

Examples - <http://java.dzone.com/articles/why-use-powermock-mock-private>

Partial support with TestNG

Able to mock final, static, and private methods

Extends EasyMock and Mockito

PowerMock is an extension of EasyMock and Mockito. It is able to mock static and private methods and classes. It is able to do this through low level implementation. It is able to mock without injecting dependencies. Since it is an extension of Mockito and EasyMock, it has their features also. PowerMock can also be used with them to mock static and private methods. Officially, it has some native support with TestNG.

Last updated 3/5/2013

## Mockito - <https://code.google.com/p/mockito/>

Examples - <http://gojko.net/2009/10/23/mockito-in-six-easy-examples/>

Used to be a fork version of EasyMock but is now its own framework

Less boilerplate code than EasyMock

Implements test spy which records method calls and actual values passed

Has a third party patch to support TestNG

Most mature and very popular

Accurate error messages

Mockito is a very popular mocking framework and it has been around for a long time. It used to be a fork version of EasyMock, but they refactored all of the code so the logic is completely different. It has a feature called test spy. It is used to record method calls and actual values passed. It is very easy to intercept method calls to inject test values. Mockito only needs one jar file to be loaded into the build path.

Last update 6/10/12

## jMockit - <https://code.google.com/p/jmockit>

Examples - <http://codebase.olsonzoo.com/2010/03/unit-testing-java-code-with-jmockit.html>

Supports TestNG

Can mock any method

Multiple ways to mock a method

Does not require interfaces to create mocks

jMockit is a very powerful mocking framework because of its flexibility. It is not a very popular mature framework. To get jMockit to work, only one jar file is needed to load into the build path. jMockit uses two types of testing, state-based and behavior-based. In state-based, stubs are used. Stub classes are created with a constructor and static methods inline with test cases. In behavior-based, the expected behavior is defined within the test cases.

Last update 3/16/2013

## jMock - <http://jmock.org/>

Examples - <http://jeantessier.com/SoftwareEngineering/Mocking.html>

Supports TestNG

Uses anonymous inner classes

Uses its own test case

Extensible API

jMock is a mature mocking framework. It is a dynamic mock object library. jMock focuses on explicitly specifying the expected behavior of the classes being tested. By design, jMock only mocks interfaces. It cannot mock static or final methods. jMock’s popularity has been decreasing and many people recommend other frameworks over jMock.

Last updated 12/19/12

## EasyMock - <https://jira.codehaus.org/browse/EASYMOCK/>

Examples - <http://jeantessier.com/SoftwareEngineering/Mocking.html>

Supports TestNG

Allows for partial mock testing

EasyMock is a mature mocking framework. It is similar to jMock in that it is also a dynamic mock object library but different in implementation. EasyMock focuses on record and replay of classes. It can verify method call order and return default values for unexpected method calls. EasyMock suffers from the loss of popularity just like jMock.

Last updated 2/16/13

Recommendation:

For its support base and popularity, Mockito is recommended.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Framework** | **Type** | **Stubbing** | **Default return values** | **Instances are strict/non-strict by default** | **Programmatic creation** | **Declarative creation** | **Injects dependencies** | **Replaces instances created using new** |
| Mockito | Proxy-based | Yes | 0, null, false | Non-strict | Yes | Yes | No | No |
| EasyMock | Proxy-based | Yes | Strict instances throw exceptions. Non-strict return 0, null, false | Depends on the ways the object was created. Type can be changed at runtime | Yes | No | No | No |
| Mockachino | Proxy-based | Yes | 0, null, false | Non-strict | Yes | No | No | No |
| jMock | Proxy-based | Yes | 0, empty String, false, empty array, null | Strict | Yes | No | No | No |
| PowerMock | Class remap | Yes | 0, null, false | Strict with EasyMock and non-strict with Mockito | Yes | No | No | Yes |
| Unitils | Proxy-based | Yes | 0, null, false | Non-strict | No | Yes | Yes | No |
| JMockit | Class remap | Yes | 0, empty collections, empty arrays, | Non-strict by default | No | Yes | Yes, replaces instances of @Mocked classes | Yes |

https://softwareinabottle.wordpress.com/2010/10/06/comparing-java-mock-frameworks-part-2-creating-mock-objects/