

[Home](#) » [Java](#) » [Core Java](#) » Mock final class

## ABOUT LUBOS KRNAC



Lubos is a Java/JavaScript developer/Tech Lead and Automation enthusiast. His religion is to constantly improve his developments skills and learn new approaches. He believes TDD drives better design and nicely decoupled code. Past experience includes C++, Assembler and C#.



## Mock final class

Posted by: Lubos Krnac | in Core Java | February 11th, 2014 | 1 Comment | 273 Views | (0 rating, 0 votes)

## Foreword

If you already read some other blog post about *unusual* mocking, you can skip prelude via this link.

I was asked to put together examples how to mock Java constructs well know for their testability issues:

- Mock private method
- Mock final method
- Mock final class
- Mock constructor
- Mock static method

I am calling these techniques *unusual* mocking. I was worried that such examples without any guidance can be widely used by teammates not deeply experienced in mocking frameworks.

Developers practicing TDD or BDD should be aware of testability problems behind these constructs and try to avoid them when designing their tests and modules. That is the reason why you probably wouldn't be facing such *unusual* mocking often on project using these great programming methodologies.

But sometimes you have to extend or maintain legacy codebase that usually contains low cohesive classes. In most cases there isn't time in current hectic agile world to make such class easy to unit test standard way. When you are trying to unit test such class you often realize that *unusual* mocking is needed.

That is why I decided to create and share refactoring considerations alongside with examples and workarounds for *unusual* mocking. Examples are using Mockito and PowerMock mocking frameworks and TestNG unit testing framework.

## Mock final class

### Refactoring considerations

Change class to non-final (remove final keyword) and test it standard way. This is technique I use always when I can change code of final class.

### Usage of PowerMock

Before usage of this example, please carefully consider if it is worth to bring bytecode manipulation risks into your project. They are gathered in **this blog post**. In my opinion it should be used only in very rare and non-avoidable cases.

## NEWSLETTER

Insiders are already enjoying weekly up-to-date complimentary whitepapers!

**Join them now** to gain **exclusive** access to the latest news in the Java world, as well as insights about Android, Scala, and other related technologies.

### Email address:



## JOIN US



With **1,240,600** unique visitors and **500** authors placed among related sites and constantly being looked out for, we encourage you. So if you have unique and interesting content then you can check out our **JCG** partners program. You can be a **guest writer** for Java Code Geek and showcase your writing skills!

2. Mocking of final void method in final class
3. Verifying of method calls in final class

Final class:

```
01 public final class Plane {
02     public static final int ENGINE_ID_RIGHT = 2;
03     public static final int ENGINE_ID_LEFT = 1;
04
05     public boolean verifyAllSystems() {
06         throw new UnsupportedOperationException("Fail if not mocked!");
07     }
08
09     public void startEngine(int engineId) {
10         throw new UnsupportedOperationException(
11             "Fail if not mocked! [engineId=" + engineId + "]");
12     }
13 }
```

Class under test:

```
01 public class Pilot {
02     private Plane plane;
03
04     public Pilot(Plane plane) {
05         this.plane = plane;
06     }
07
08     public boolean readyForFlight() {
09         plane.startEngine(Plane.ENGINE_ID_LEFT);
10         plane.startEngine(Plane.ENGINE_ID_RIGHT);
11         return plane.verifyAllSystems();
12     }
13 }
```

Test:

```
01 @PrepareForTest(Plane.class)
02 public class PilotTest extends PowerMockTestCase {
03     @Test
04     public void testReadyForFlight() {
05         Plane planeMock = PowerMockito.mock(Plane.class);
06         Pilot pilot = new Pilot(planeMock);
07
08         Mockito.when(planeMock.verifyAllSystems()).thenReturn(true);
09
10         // testing method
11         boolean actualStatus = pilot.readyForFlight();
12
13         Assert.assertEquals(actualStatus, true);
14         Mockito.verify(planeMock).startEngine(Plane.ENGINE_ID_LEFT);
15         Mockito.verify(planeMock).startEngine(Plane.ENGINE_ID_RIGHT);
16     }
17 }
```

## Links

- Source code can be downloaded from Github.

Other *unusual* mocking examples:

- Mock private method
- Mock final method
- Mock constructor
- Mock static method

**Reference:** Mock final class from our JCG partner Lubos Krnac at the <http://lkrnac.net/> blog.

Tagged with: MOCKITO TESTING



(0 rating, 0 votes)

You need to be a registered member to rate this. 1 Comment 273 Views Tweet it!

Do you want to know how to develop your skillset to become a **Java Rockstar?**

Subscribe to our newsletter to start Rocking right now!  
To get you started we give you our best selling eBooks for **FREE!**