

ToiletPaper #149

Testing with Mockito

Autor: Alexander Pöhlmann / Software Engineer / Standort Leipzig

✗ Problem

Mockito ist, mit seinen Möglichkeiten Klassen zu mocken, ein sehr mächtiges Tool. Leider kommt auch Mockito mit seinen *verify()*-Methoden an seine Grenzen, wenn die *equals()*-Methoden der zu prüfenden Objekten nicht existieren oder fehlerbehaftet sind. Im Normalfall werden die Klassen einfach behoben und es hat sich erledigt. Im Falle von externen Klassen wird es nicht so einfach. Hier gibt es zum Beispiel nur die Möglichkeit, diese in eigene Klassen zu überführen, sehr gut in einem Wrapper zu verstecken oder gleich zu Kotlin zu wechseln.

✓ Lösung

Eine brauchbare Lösung, um mit dem Test den Code vollständig abzudecken, ist die Verwendung von *ArgumentCapture*. Hiermit ist es möglich, die Instanzen beim *verify()* abzufangen und im späteren Zeitpunkt zu prüfen.

➔ Beispiel

```
1 FuelPortion expectedFuelPortion = FuelPortion.DYNAMIC;
2
3 Engine myMockEngine = mock(Engine.class); //Or with @Mock
4 Car myTestCar = new Car(myMockEngine);
5
6 myTestCar.startEngine();
7
8 ArgumentCaptor<EngineConfiguration> engineConfigurationCapture = ArgumentCaptor.forClass(EngineConfiguration.class); //Or with @Captor
9 verify(myMockEngine).start(engineConfigurationCapture.capture()); // capture() is the central call for ArgumentCapture
10
11 EngineConfiguration actualEngineConfiguration = engineConfigurationCapture.getValue();
12 Assert.assertEquals(expectedFuelPortion, actualEngineConfiguration.getFuelPortion());
```

+ Weiterführende Aspekte

- <https://www.vogella.com/tutorials/Mockito/article.html>
- <https://www.baeldung.com/mockito-series>
- <https://github.com/mockito/mockito>