

Sprytne testowanie ze Spock Framework

Część I: Krótkie wprowadzenie

O mnie

Łukasz Krauzowicz

Programista,
Miłośnik ekosystemu JVM,
Entuzjasta Otwartego Oprogramowania

lukasz.krauzowicz@ailleron.com

<https://github.com/kraluk>

Agenda

Agenda

- Spock Framework? A co to?
 - Struktura
 - Pola
 - Rozszerzenia
 - Mockowanie
 - Power Assertion
- Hands-on coding - czyli jak testować w Spocku
- Spock vs JUnit - formalnie



Spock Framework



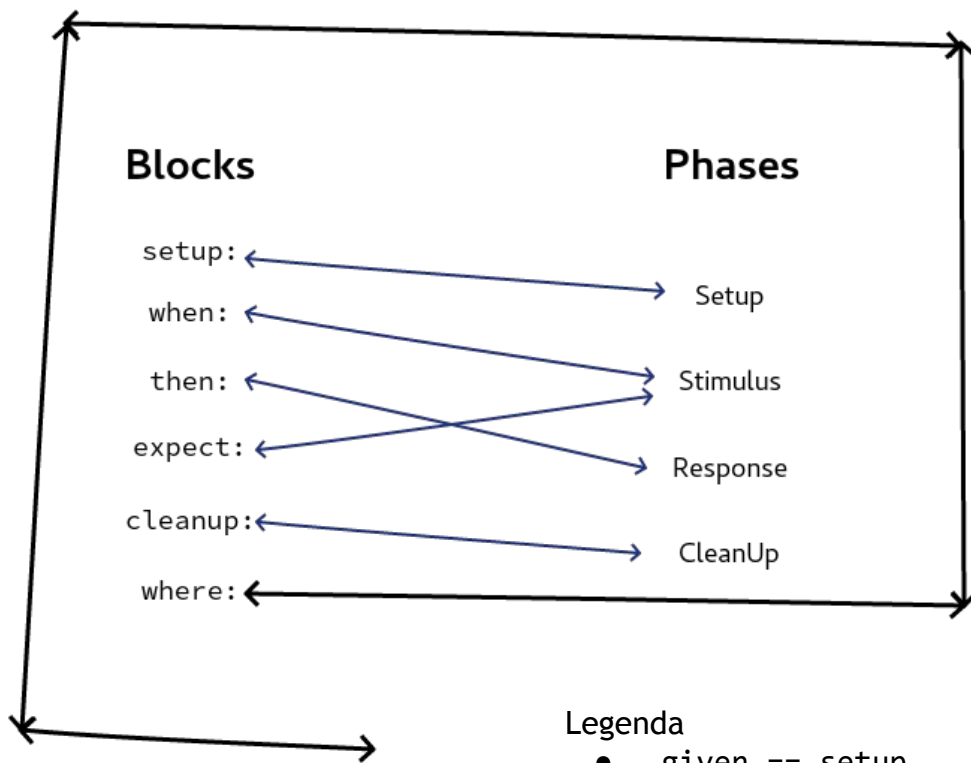
Spock Framework

- *“Spock is a testing and specification framework for Java and Groovy applications”*
- Tylko Java i Groovy :(
- BDD - Behavioural Driven Development
- Aktualna wersja: **1.1-groovy-2.4**
- Strona domowa: <http://spockframework.org/>
- Konsola webowa: <https://meetspock.appspot.com/>

```
import spock.lang.*
```

Struktura ~~testu~~ specyfikacji

Bloki i fazy w Spocku



Pola

Pola

- Reinicjowane po każdym teście
- `@Shared def first = 1.0`
- `static def second = 2.0`

Rozszerzenia

Rozszerzenia (Extensions)

- `@Timeout` timeout dla danego testu
- `@Ignore` to co tygryski lubią najbardziej :)
- `@IgnoreRest` ignoruj całą resztę
- `@FailsWith` Analogiczne dla junitowego
`@Test(expected = ...)`, ale nie do końca!

Hands-on coding!

Mockowanie

Mockowanie

```
Random random = Mock(Random)  
random.nextDouble() >> 1.0
```

albo...

```
Random random = Mock() {  
    1 * nextDouble() >> 1.0  
}
```


Power Assertion

Power Assertion

```
assert [[1,2,3,3,3,3,4]].first().unique() == [1,2,3]
```

```
// Assertion failed:
```

```
// assert [[1,2,3,3,3,3,4]].first().unique() == [1,2,3]
```

```
//
//
//
//
```

A jUnit?

Spock vs jUnit

Spock	jUnit
Specification class	Test class
setup()	@Before
cleanup()	@After
setupSpec()	@BeforeClass
cleanupSpec()	@AfterClass
Feature	Test
Condition	Assertion
Exception condition	@Test(expected = Exception.class)
Interaction	Mock expectation

Materialy



spockframework.org/spock/docs/1.1/spock_primer.html

Keyword: **spock 1.1 primer**



github.com/kraluk/smart-spock

Podsumowanie

Podsumowanie

- Groovy
- Specification
- **given/when/then**
- Specification as Documentation
- Power Assertions
- Specification :)



Dzięki za uwagę!

