WYKORZYSTANIE JĘZYKA GROOVY W TESTACH JEDNOSTKOWYCH, INTEGRACYJNYCH I AUTOMATYCZNYCH

Mirosław Gołda,
Programista Java

Agenda

- Kilka słów o języku Groovy
- Wpięcie Grooviego w projekt Springowy
- Testy jednostkowe w Groovy
- Testy integracyjne w Groovy
- Testy automatyczne w Groovy
- Jakieś niedogodności?

Groovy - cechy

- Język skryptowy, dynamiczny, działający na platformie JVM
- Składnia podobna do Javy, niski próg wejścia w język
- Wiele elementów składni upraszczających tworzenie kodu
- Możliwość statycznej kompilacji i sprawdzania typów
- Świetna i bezproblemowa integracja z Javą

Prosty przykład 1

Poprawny kod Groovy i Java:

```
public class HelloWorld {
  public static void main(String[] args) {
    for (int i=1; i<=5; i++) {
      System.out.println("Hello world no." + i);
    }
  }
}
```

Możemy go jednak uprościć i uczynić bardziej Groovy.

Prosty przykład 2

Ale można też tak…

```
(1..5).each {
  println "Hello world no.${it}"
}
```

Skrypt Grooviego bez jawnej deklaracji klasy.

Groovy – nowe możliwości

Kilka z listy uproszczeń

- Brak średników (można ich używać)
- Możliwość pominięcia deklaracji typu za pomocą `def`
- Domyślny modyfikator widoczności `public`
- Wzbogacone możliwości pracy ze Stringami (GStrings)
- Dodatkowe możliwości pracy na kolekcjach
- Groovy Beans
- Brak potrzeby jawnej deklaracji `return`
- Domknięcia dostępne chyba od zawsze
- …

Część z tych udogodnień za chwilę zobaczymy.

Włączenie Groovy w projekcie Java

W build.gradle dodajemy:

```
...
apply plugin: "groovy"
...
dependencies {
...
  compile 'org.codehaus.groovy:groovy-all:2.4.1'
...
}
```

Zmieniamy w strukturze plików:

`src/java ➔ src/groovy`

Spock framework – testy jednostkowe i integracyjne

Czemu jest fajny:

- jasny podział na bloki **given**/**when**/**then**,
- czytelne nazwy testów,
- czytelne raporty niepowodzenia testów,
- ciekawe konstrukcje, np. tabele w bloku **where**,
- Groovy ☺… a więc szansa na zwięzły i czytelny kod testów.

Spock framework - instalacja

Dodajemy do build.gradle:

```
dependencies {
...
    testCompile 'org.spockframework:spock-core:0.7-groovy-2.0'
    testCompile 'org.spockframework:spock-spring:0.7-groovy-2.0'
    testCompile 'org.springframework:spring-test:4.1.5.RELEASE'
    testCompile 'cglib:cglib-nodep:3.1'
...
}
```

Spock framework – najprostszy test

```
class MyFirstSpec extends Specification {
  def "should add up to 5 when 2 and 2
given!"() {
    expect:
    2 + 2 == 5
  }
}
```

Spock framework – wynik testu

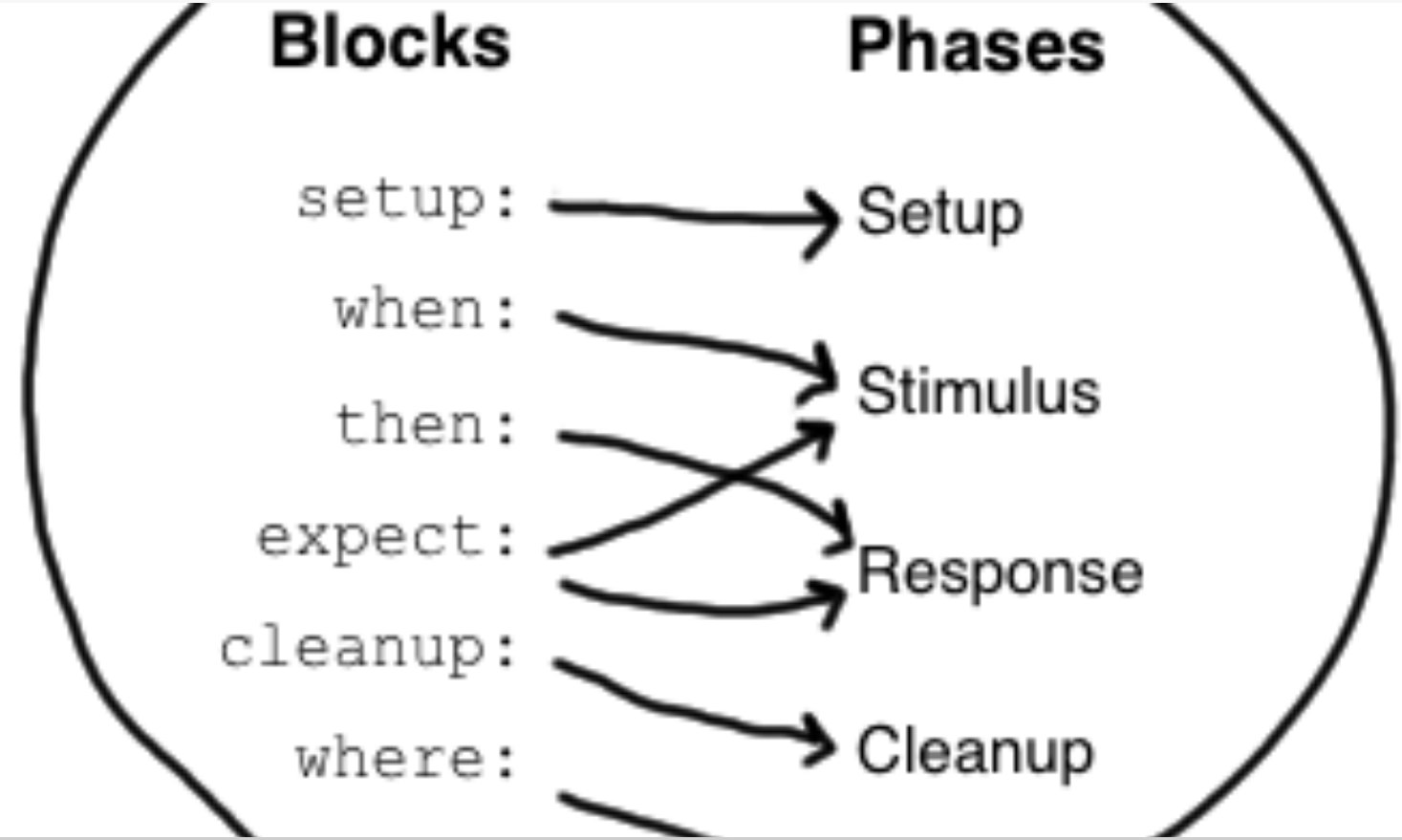
```
MyFirstSpec
 - should add up to 5 when 2 and 2 given!   FAILED

   Condition not satisfied:

   2 + 2 == 5
     |   |
     4   false

   at MyFirstSpec.should add up to 5 when 2 and 2 given!
(Script1.groovy:7)
```

Spock framework – dostępne bloki



https://code.google.com/p/spock/wiki/SpockBasics

Spock vs JUnit

Spock	JUnit
Specification	Test class
`setup()`	`@Before`
`cleanup()`	`@After`
`setupSpec()`	`@BeforeClass`
`cleanupSpec()`	`@AfterClass`
Feature	Test
Parametrized feature	Theory
Condition	Assertion
Exception condition	`@Test(expected=…)`
`@FailsWith`	`@Test(expected=…)`
Interaction	Mock expectation (EasyMock, JMock…)

https://code.google.com/p/spock/wiki/SpockBasics

Groovy weather

Co robimy:

- Uruchamiamy aplikację
- Krótki przegląd aplikacji (struktura, gradle)
- Testy jednostkowe (kontroler, serwis)
- Testy integracyjne (serwis, kontroler)
- Testy automatyczne (gradle, config, Spec, Page, Module)

Ciemne strony korzystania z języka Groovy I Spocka…

- Gorsze wsparcie IDE niż w przypadku Javy
- Dynamiczne typowanie sprawia, że nie wszystkie błędy można wyłapać już na etapie kompilacji
- **"With great power comes great responsibility" – Voltaire**

  Dotyczy to niestety także pisania w Groovim, można w nim stworzyć nieczytelny kod…
  Ale w jakim języku nie można?
- Wsparcie za strony Pivotala dla rozwoju projektów Groovy i Grails kończy się 31 marca 2015 r. Ale prężna społeczność open source.

Linki

- http://groovy-lang.org/
- http://www.gradle.org/docs/current/userguide/groovy_plugin.html
- http://spockframework.org/
- http://www.gebish.org/
- https://github.com/geb/geb-example-gradle
- http://meetspock.appspot.com/
- https://github.com/mirog/groovy-testing-example/

Wykorzystanie języka Groovy w testach jednostkowych, integracyjnych i automatycznych

Dziękuję

Mirosław Gołda

Programista Java

e-mail: miroslaw.golda@gmail.com