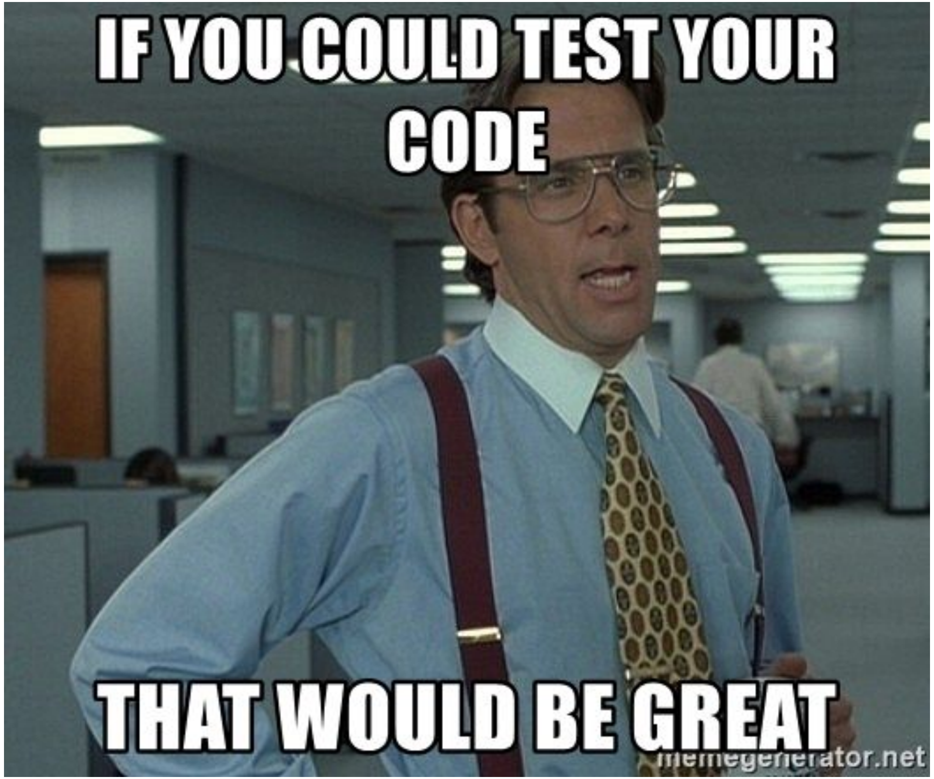
Test con Groovy

Aplicando test a historias de usuarios.


IF YOU COULD TEST YOUR CODE
THAT WOULD BE GREAT
memegenerator.net

¿Qué tipos de test encontramos en el Bash Core?

- Test en java y groovy
- Junit y Spock

¿Qué es Spock Framework?

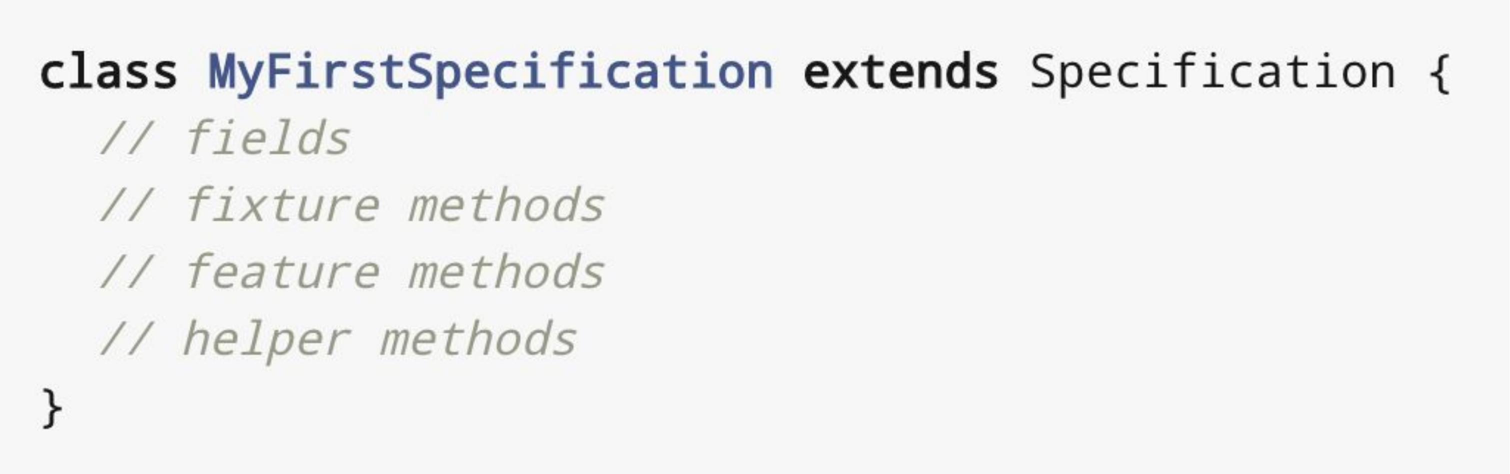
Spock es un marco de prueba y especificaciones para aplicaciones Java y Groovy.

Terminología

- System Under Specification (SUS)
- Specification
- Fields
- Fixture Methods
- Feature Methods
- Blocks

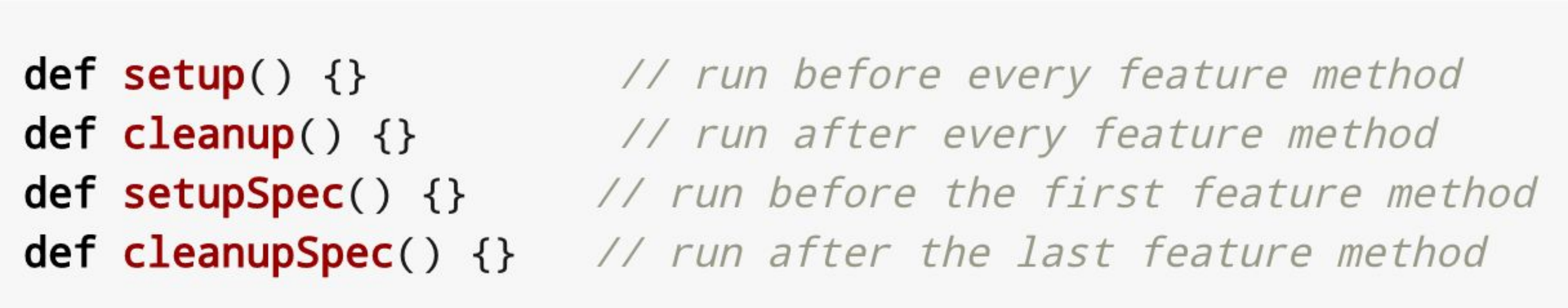
Specification

- Package **spock.lang** contains the most important types for writing specifications.
- JUnit helps to run Spock specification.



```
class MyFirstSpecification extends Specification {
  // fields
  // fixture methods
  // feature methods
  // helper methods
}
```

Fixture Methods



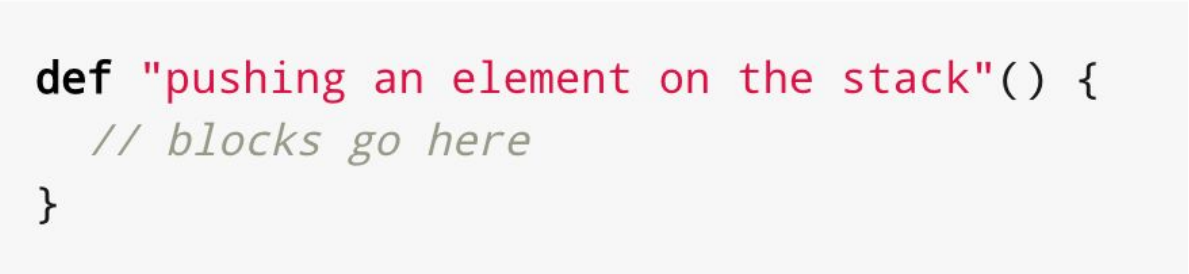
```
def setup() {}          // run before every feature method
def cleanup() {}        // run after every feature method
def setupSpec() {}     // run before the first feature method
def cleanupSpec() {}   // run after the last feature method
```

**** Todos los fixture methods son opcionales.***

Feature Methods

Conceptually, a feature method consists of four phases:

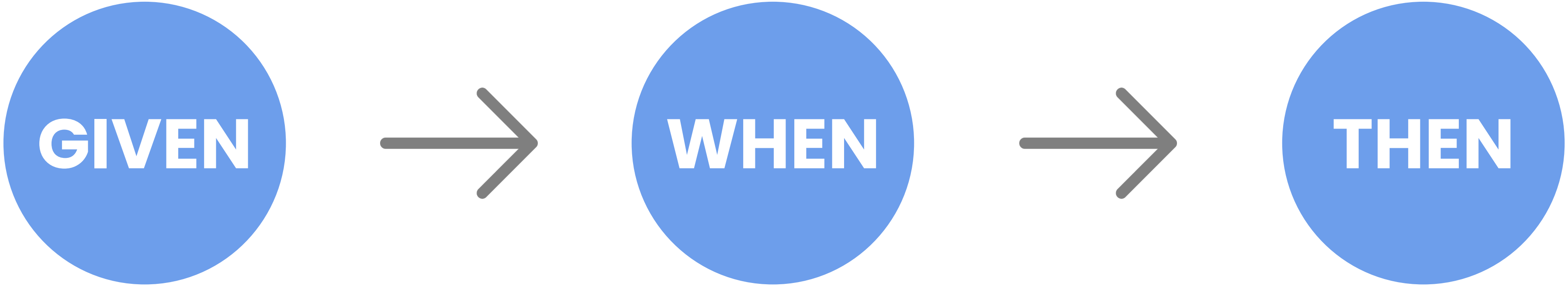
1. Set up the feature’s fixture
2. Provide a stimulus to the system under specification
3. Describe the response expected from the system
4. Clean up the feature’s fixture



```
def "pushing an element on the stack"() {
  // blocks go here
}
```

SPOCK FRAMEWORK

BDD



GIVEN → WHEN → THEN

TALK IS CHEAP
SHOW ME THE CODE
imgflip.com

User Story

Description: Validar el resultado de la suma de dos números enteros.

- **Given:** Los dos números enteros son 4 y 7.
- **When:** Estos dos números se deben sumar.
- **Then:** El resultado tiene que ser 11.

SPOCK FRAMEWORK

User Story

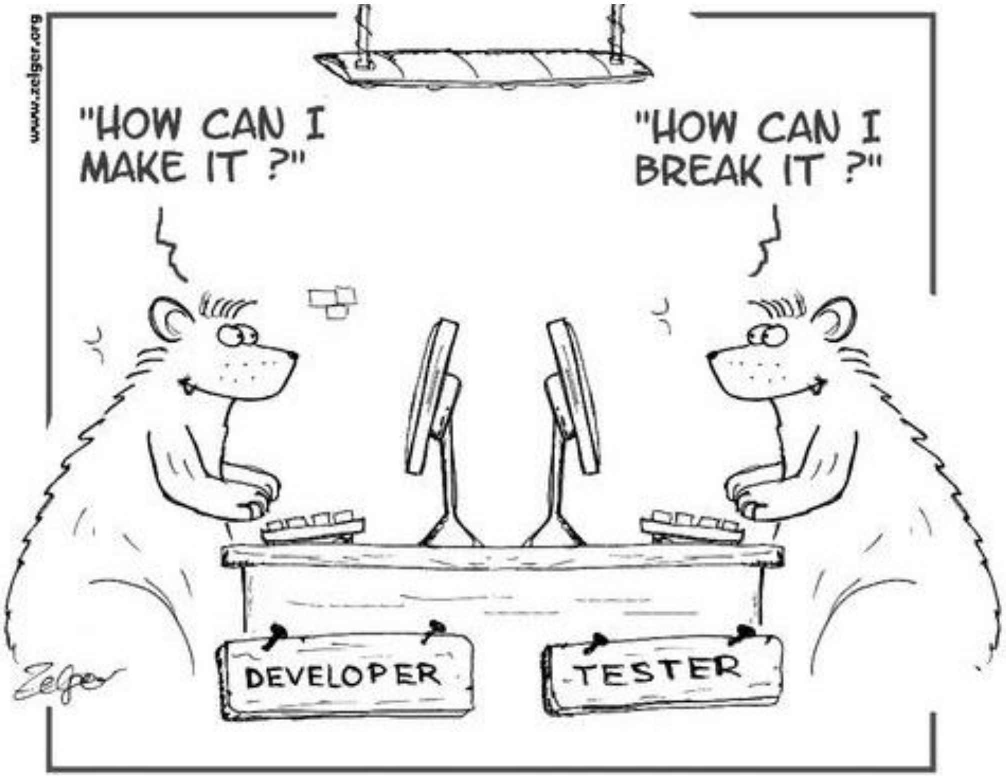
Description: Cuando se agrega un elemento a una lista, el tamaño debe cambiar.

- **Given:** Una lista de 4 números.
- **When:** Agregar un nuevo elemento a la lista.
- **Then:** El nuevo tamaño de la lista debe ser 5 y el antiguo tamaño de la lista debe ser 4.

User Story

Description: Quiero saber si un número es par.

- **Given:** Dado un número entero.
- **When:** Obtener el resto del número entero.
- **Then:** El resultado debe ser igual a cero indicando que es un número par.

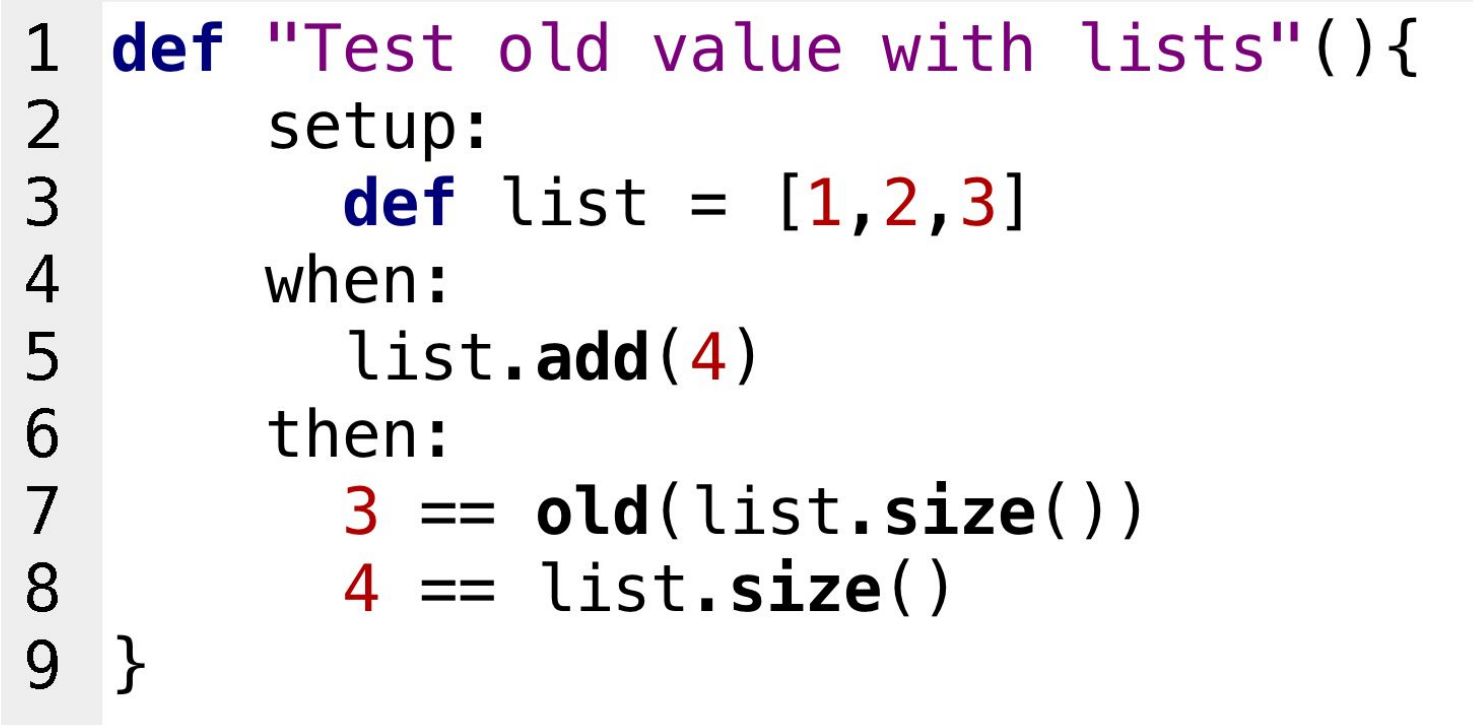

www.zelger.org
"HOW CAN I MAKE IT ?"
"HOW CAN I BREAK IT ?"
DEVELOPER
TESTER
Zelger

SPOCK FRAMEWORK

Blocks

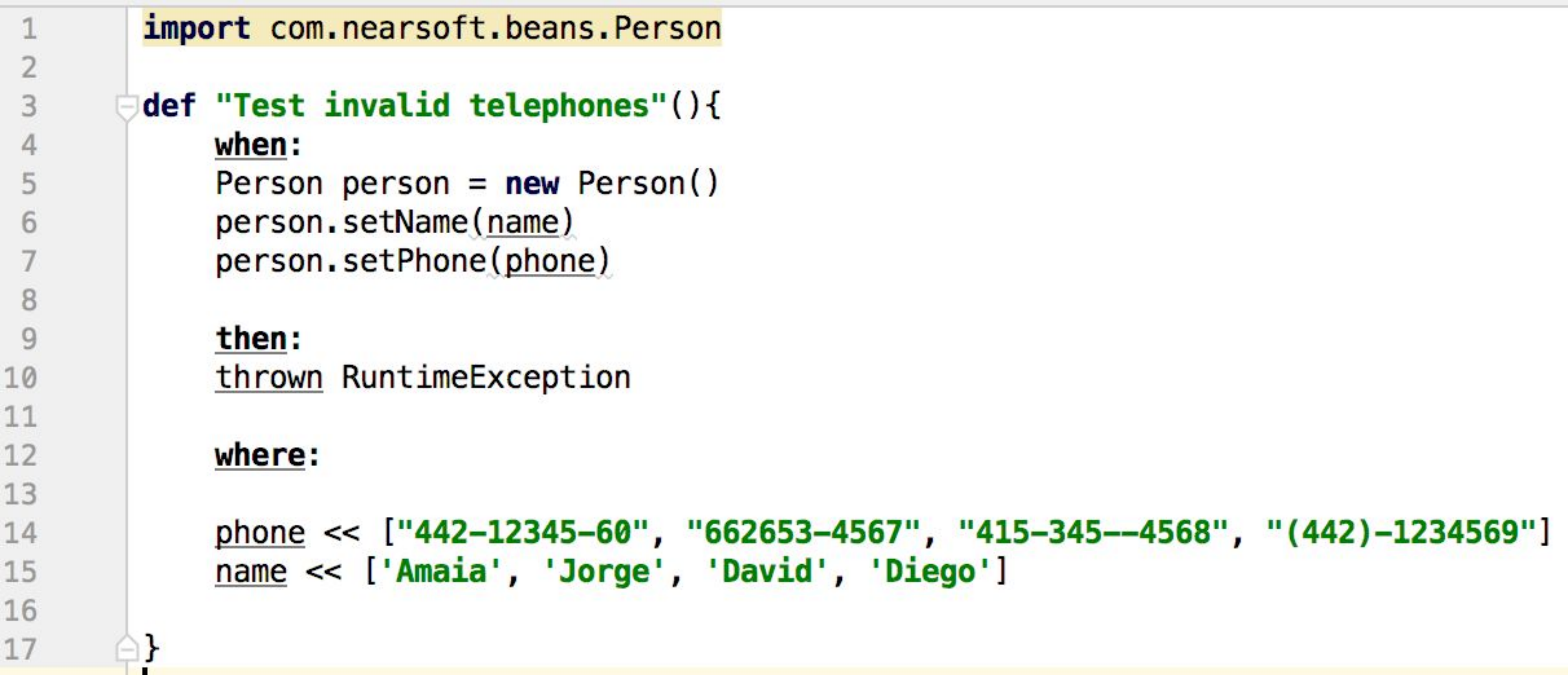
Block name	Description
given	preconditions, data fixtures
when	actions that trigger some outcome
then	makes assertions about outcome
expect	short alt to when & then
where	applies varied inputs
and	subdivides other blocks
setup	alias for given
cleanup	post-conditions, housekeeping

Blocks



```
def "Test old value with lists"(){
    setup:
      def list = [1,2,3]
    when:
      list.add(4)
    then:
      3 == old(list.size())
      4 == list.size()
}
```

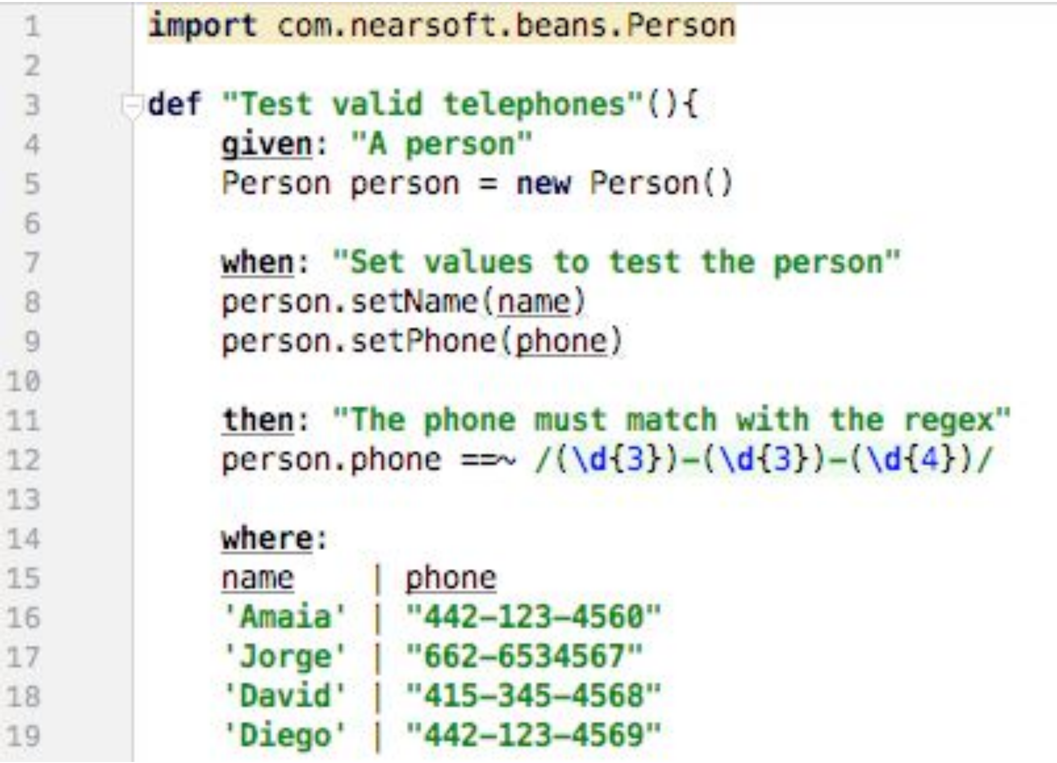
Data Pipes



```
import com.nearsoft.beans.Person

def "Test invalid telephones"(){
    when:
    Person person = new Person()
    person.setName(name)
    person.setPhone(phone)

    then:
    thrown RuntimeException

    where:

    phone << ["442-12345-60", "662653-4567", "415-345--4568", "(442)-1234569"]
    name << ['Amaia', 'Jorge', 'David', 'Diego']

}
```

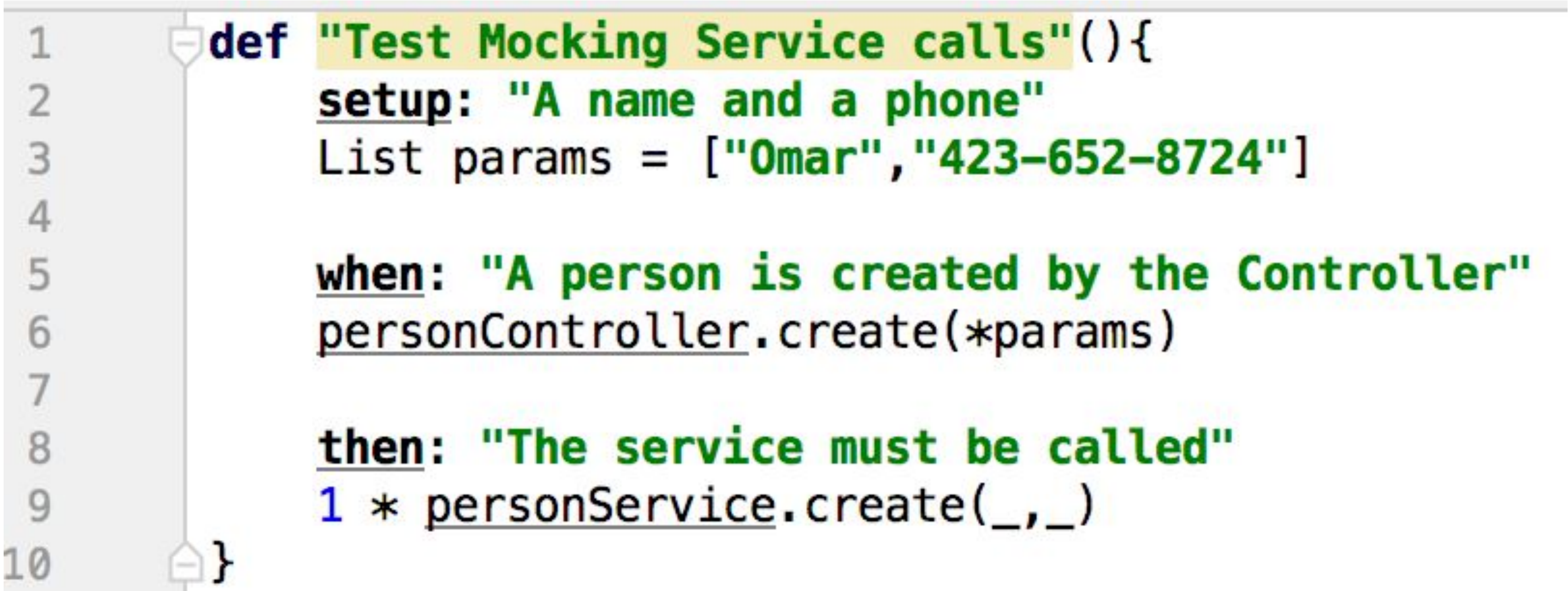
SPOCK FRAMEWORK

Data Tables



```
import com.nearsoft.beans.Person

def "Test valid telephones"(){
    given: "A person"
    Person person = new Person()

    when: "Set values to test the person"
    person.setName(name)
    person.setPhone(phone)

    then: "The phone must match with the regex"
    person.phone ==~ /(\d{3})-(\d{3})-(\d{4})/

    where:
    name    | phone
    'Amaia' | "442-123-4560"
    'Jorge' | "662-6534567"
    'David' | "415-345-4568"
    'Diego' | "442-123-4569"
```

Interactions



```
def "Test Mocking Service calls"(){
    setup: "A name and a phone"
    List params = ["Omar","423-652-8724"]

    when: "A person is created by the Controller"
    personController.create(*params)

    then: "The service must be called"
    1 * personService.create(_,_)
}
```

SPOCK FRAMEWORK

Documentación

- http://spockframework.org/
- https://www.innuy.com/blog/what-is-bdd-behavior-driven-development/

¿Preguntas?

Muchas gracias