Unit Testing

With Spock Framework

Follow along with your favorite tool:

JUnit + Mockito, EasyMock, JMock, etc. or

MSUnit + Moq, etc.

Setup – Pretend File System

- Write a class File
  - Add size property
- Write a class Folder with following functionalities
  - We can add Files to a Folder
  - We can add other Folders to a Folder

Value based unit-testing

- Implement size method in the folder sums up all the sizes of fi folders
  - Test Driven Development (TDD), or Not
  - What test-cases?

Rec: Unit class per method under test

- Not, unit class per class under test
- Example: FolderSizeTest

Rec: Use Given, When, Then format

- Example: m.FolderSizeTest.whenMoreThanOneFile()
- Where are the assertions?
  - Implicit
  - Helpers methods will need explicit assertions

Data Tables

- Example: m.FolderSizeTest.twoFiles
- Unroll can help with debugging
- There are other ways to write data tables

Interaction based unit-testing

- Implement Shortcut class
  - We can have Shortcut to a File, Folder, or another Shortcut
  - We can add Shortcut to a Folder
  - Implement open method in Shortcut that opens the underlying
  - TDD or Not?
  - What test cases?

Mock

- Mock the interactions:
  - Example: m.ShortcutOpenTest.setup()
- Verify the expectations:
  - Example: m.ShortcutOpenTest.openingShortcutOpensFile()

Mock Vs. Stub

- Everyone has their own definition
- Every API has their own definition
- In Spock:
  - To stub is making collaborators respond
  - Mock() can be used for mocking and stubbing
  - Stub() can only be used for stubbing
  - Example: m.FolderSizeStubTest

Summary

- Write a unit test class per method under test
- Using TDD preferably
- Leverage the popular given, when, then to structure tests
- Use data table concept to reduce duplicate code
- Employ mocks to perform interaction based unit-testing betw
  collaborators

References

- Spock Framework (http://spockframework.github.io/spock/docs/1.0/spock_primer.html)

Advanced Topics

- Exceptions
  - Thrown
  - Not thrown
- Lenient Vs. Strict
  - Spock Framework, Mockito are lenient
  - EasyMock, JMock are strict
- Spy and Partial Mock
  - Spy()
    - Delegate to the real object
    - But, enforce the collaborations
    - Use carefully!
  - Partial Stubs (using Spy())
    - Stub some methods
    - Delegate the rest to the real object
    - Use carefully!