Advanced Topics

Nicolae Caprarescu
FULL-STACK SOFTWARE DEVELOPMENT CONSULTANT

www.properjava.com

Overview

Testing code that deals with time

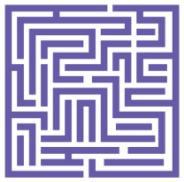
The EasyMockSupport class

Using partial mocks

Mocking final methods

Testing Code That Deals with Time

One of the most frequently encountered difficult testing contexts



Solving this situation typically involves freezing time

We'll explore two ways of writing tests for this context

Demo

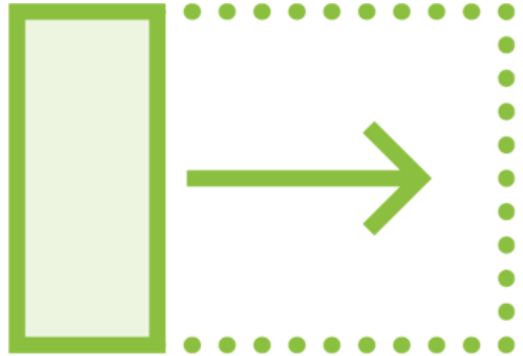
Testing the AccountClosingService

Solution One: The PowerMock Framework

EasyMock cannot mock static methods, such as LocalDate.now()

This solution is useful when you *can't* change the source code making static calls

https://github.com/powermock/powermock/wiki/EasyMock

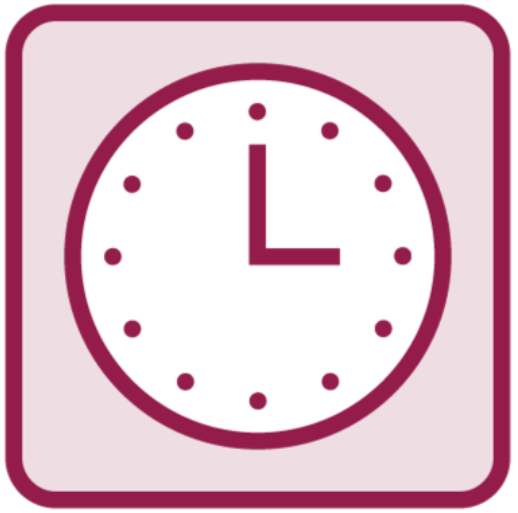


We won't cover PowerMock in this course

Solution Two: Java's Clock Class

Pluggable abstraction of time which simplifies tests

This solution is useful when you *can* change the source code making static calls



https://docs.oracle.com/javase/8/docs/api/java/time/Clock.html (since Java 1.8)

No need for any other additional frameworks

Demo

Back to finishing the AccountClosingServiceTest

The EasyMockSupport Class

Helps you manage your mocks

Turns replay(...mocks...)/verify(...mocks...) into replayAll()/verifyAll()

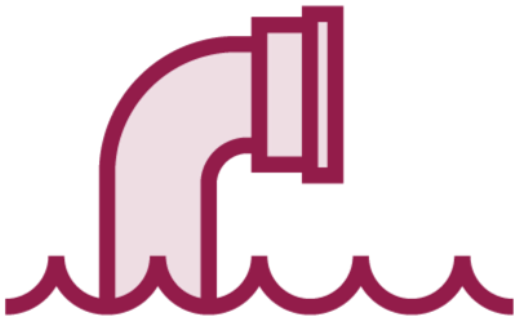
To use it, you can extend it or delegate to it

Demo

Using EasyMockSupport in the AccountOpeningServiceTest

Partial Mocks in EasyMock

Useful when you only want to mock a subset of the methods in a class



Also called spies

Use occasionally

Consider refactoring before using

Demo

Using partial mocks to test the ExternalInvestmentManagementService

Partial mocks should only be used occasionally and come in useful when you find yourself in a situation with code that impossible, or too costly, to change

Mocking Final Methods with EasyMock

Mocking final classes and methods is not supported

You have to use PowerMock, which extends EasyMock

We don't cover PowerMock in this course

Summary

Gained the ability to test code that deals with time:

- PowerMock
- Java's Clock API

Used EasyMockSupport to help you keep track of mocks

Used partial mocks to test the ExternalInvestmentManagementService

Discussed mocking final methods

Course Summary

- Why use mocking, test doubles, and creating mocks
- Configuring mocked methods and using argument matchers
- Verifying that the expected mocked methods are used exactly as described during actual test execution
- Testing code that deals with time and using partial mocks to test code that is difficult to refactor