Japanese Using EasyMock with Groovy

EasyMock is a mocking framework for Java. Here we look at EasyMock 2.2 which requires Java 5 and has the following benefits:

- Hand-writing classes for Mock Objects is not needed.
- Supports refactoring-safe Mock Objects: test code will not break at runtime when renaming methods or reordering method parameters
- Supports return values and exceptions.
- Supports checking the order of method calls, for one or more Mock Objects.

The sections below illustrate using EasyMock for the mocking parts of Using Testing Frameworks with Groovy.

The Item Storer Example

We are going to consider how you might use EasyMock as part of testing the Item Storer Example.

Here is how we can test `JavaStorer`:

```
// require(groupId:'easymock', artifactId:'easymock', version='2.2')
import org.easymock.EasyMock

mockControl = EasyMock.createStrictControl()
mockReverser = mockControl.createMock(Reverser.class)
storer = new JavaStorer(mockReverser)
testStorage()

def testStorage() {
    expectReverse(123.456, -123.456)
    expectReverse('hello', 'olleh')
    mockControl.replay()
    checkReverse(123.456, -123.456)
    checkReverse('hello', 'olleh')
    mockControl.verify()
}

def expectReverse(input, output) {
    // it's a pity mockControl doesn't have an expect() method
    EasyMock.expect(mockReverser.reverse(input)).andReturn(output)
}

def checkReverse(value, reverseValue) {
    storer.put(value)
    assert value == storer.get()
    assert reverseValue == storer.getReverse()
}
```