Recitation 2: Testing with Stubs

Venkata Nikitha Machineni
Paulo Canelas

Carnegie Mellon University

What is Unit Testing?

- Test individual units/components of system
- Not test functionality of dependencies
  - Trust implementation
  - Will test it ourselves
  - No implementation yet

Example 1

1. Suppose we want to test a **AutonomousCar** object driving from two locations;

2. Consider a **Route** class that uses a complex search algorithm to find the shortest path between two GPS locations;

3. The algorithm is **slow**;

4. Just need to give some directions to the **AutonomousCar** object.

Example 2

1. **□oTController** unlocks doors when someone has arrived to the house;

2. **Cannot actually go** to different locations;

3. Just need geolocation information to test algorithm.

Terminology

- Dummy
- Fake
- Stub
- Mock

Fake Objects

Optimized and stripped-down working implementations of some functionality;

+ **Less overhead**
- **Not production ready**

Fake Example

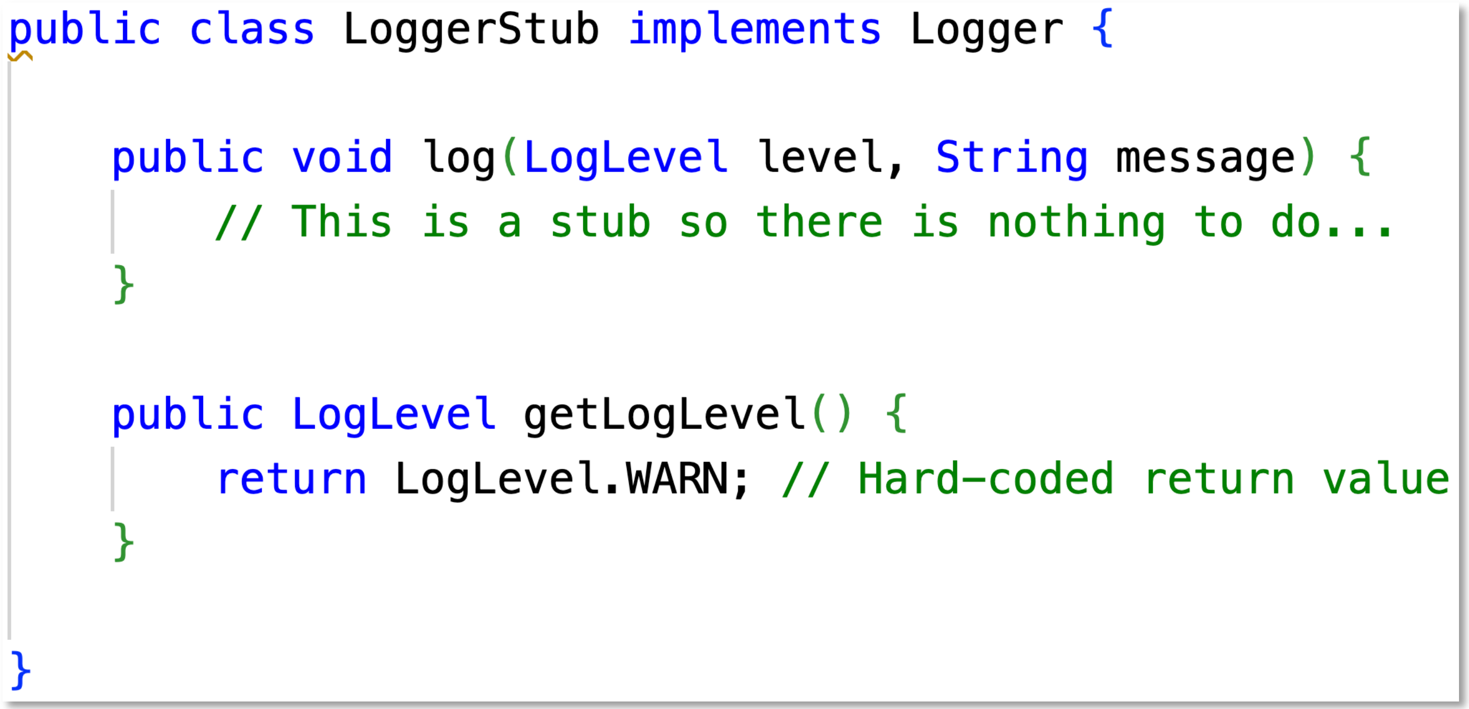
```
public class FakeUserRepository implements UserRepository {

    private Collection<User> users = new ArrayList<User>();

    public void save(User user) {
        if (findById(user.getId()) == null)
            users.add(user);
    }

    public User findById(String id) {
        for (User user : users) {
            if (user.getId().equals(id))
                return user;
        }
        return null;
    }
}
```

Stub Objects

Holds predefined data

Answers calls during tests

Stub Objects

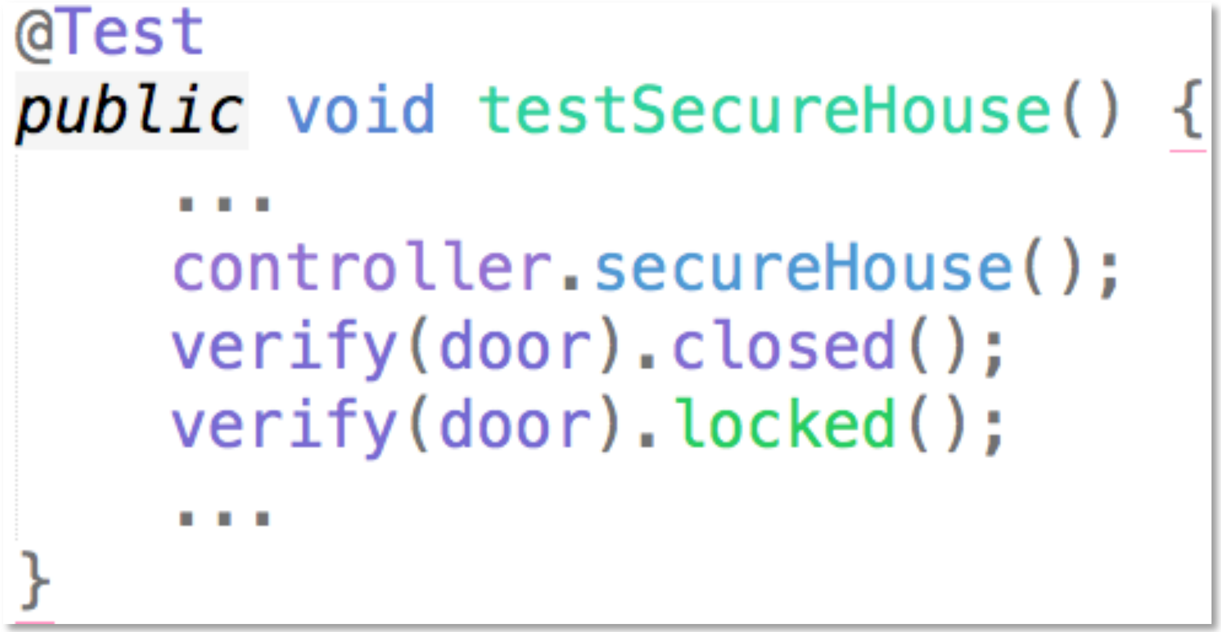


```
public class LoggerStub implements Logger {

    public void log(LogLevel level, String message) {
        // This is a stub so there is nothing to do...
    }

    public LogLevel getLogLevel() {
        return LogLevel.WARN; // Hard-coded return value
    }

}
```

Mock Objects

- Register calls they receive
- Assert/Verify behavior
- Do not return values

Mock Example



```
@Test
public void testSecureHouse() {
    ...
    controller.secureHouse();
    verify(door).closed();
    verify(door).locked();
    ...
}
```

Advantages and Disadvantages

+ Test object interactions
+ Encourage modular design
+ Testing unimplemented dependencies

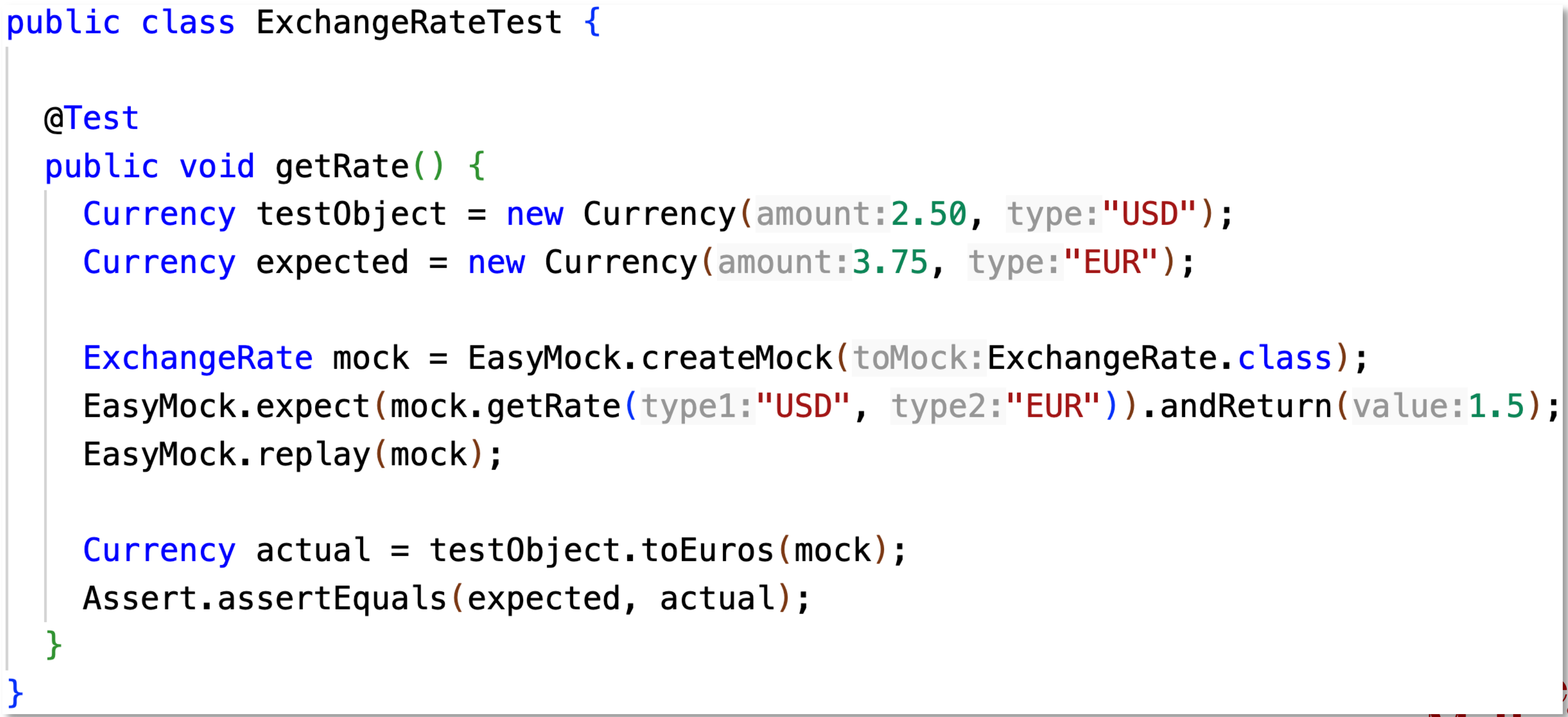
- Code complexity
- More code to maintain
- Not integration testing
- Testing behavior, not other attributes (e.g., performance)

Mocking Java Frameworks

EasyMock

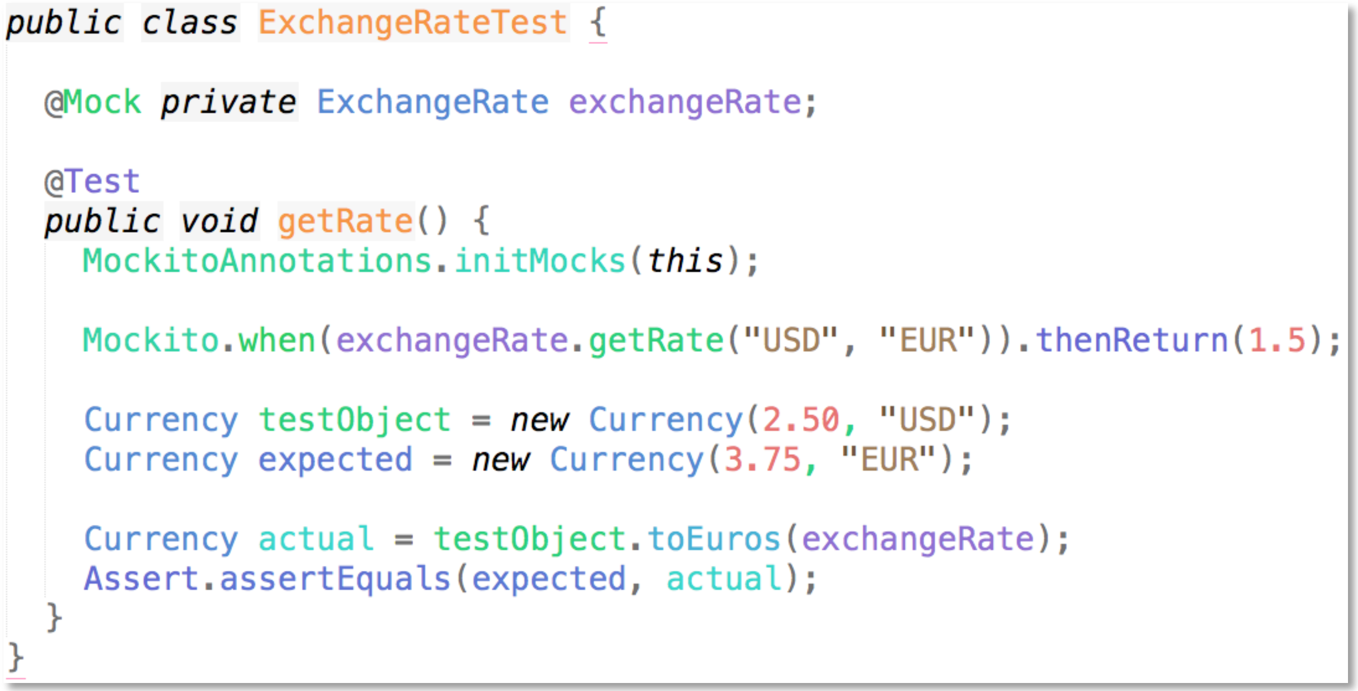
Mockito

EasyMock Example



```
public class ExchangeRateTest {

  @Test
  public void getRate() {
    Currency testObject = new Currency(amount:2.50, type:"USD");
    Currency expected = new Currency(amount:3.75, type:"EUR");

    ExchangeRate mock = EasyMock.createMock(toMock:ExchangeRate.class);
    EasyMock.expect(mock.getRate(type1:"USD", type2:"EUR")).andReturn(value:1.5);
    EasyMock.replay(mock);

    Currency actual = testObject.toEuros(mock);
    Assert.assertEquals(expected, actual);
  }
}
```

Mockito Example



```
public class ExchangeRateTest {

  @Mock private ExchangeRate exchangeRate;

  @Test
  public void getRate() {
    MockitoAnnotations.initMocks(this);

    Mockito.when(exchangeRate.getRate("USD", "EUR")).thenReturn(1.5);

    Currency testObject = new Currency(2.50, "USD");
    Currency expected = new Currency(3.75, "EUR");

    Currency actual = testObject.toEuros(exchangeRate);
    Assert.assertEquals(expected, actual);
  }
}
```

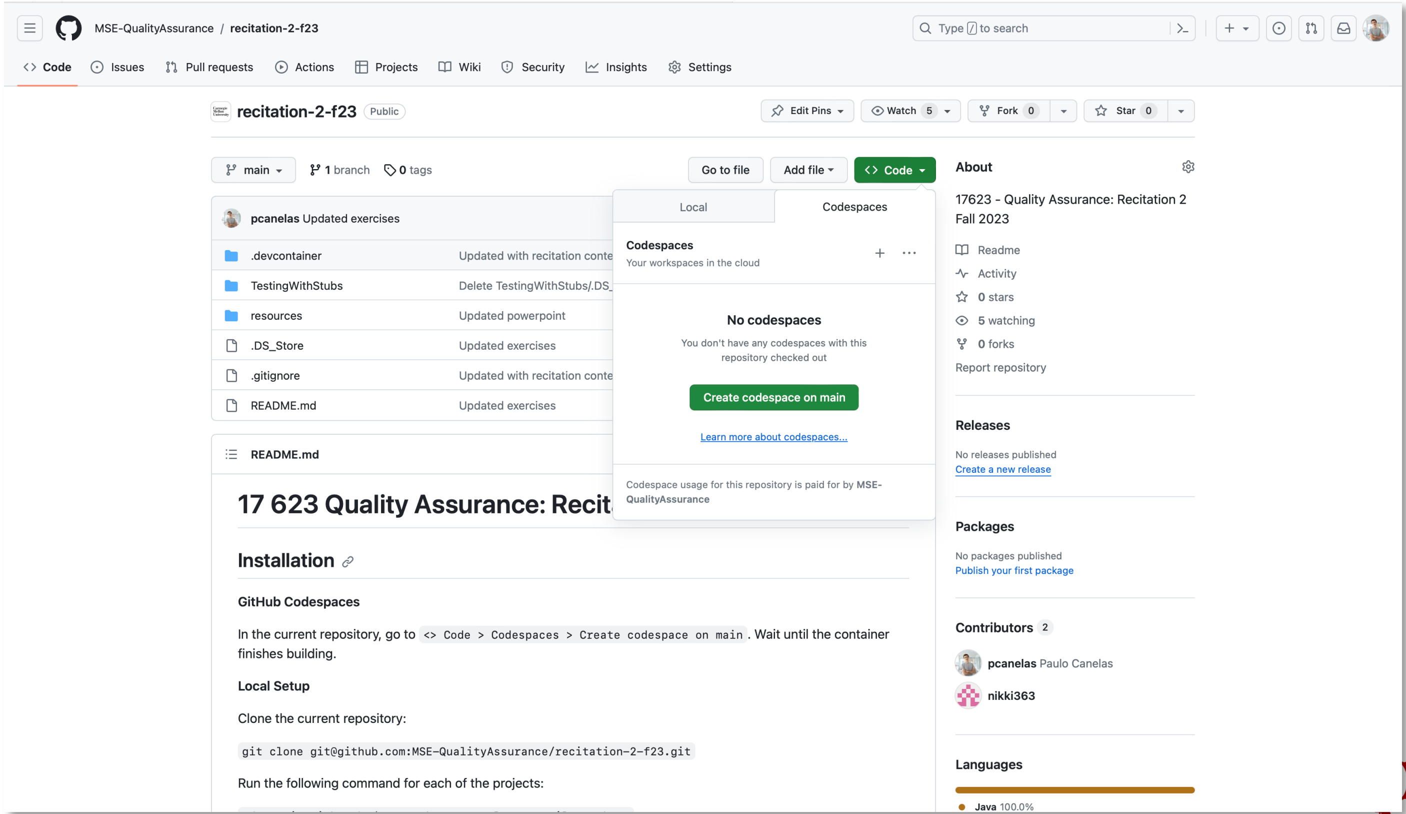
Carnegie Mellon University

Exercise

1. Implement **toDollars** method in Currency

2. Use your favorite mocking framework

https://github.com/MSE-QualityAssurance/recitation-2-f23



pcanelas Updated exercises

.devcontainer	Updated with recitation conte
TestingWithStubs	Delete TestingWithStubs/.DS_
resources	Updated powerpoint
.DS_Store	Updated exercises
.gitignore	Updated with recitation conte
README.md	Updated exercises

Codespaces
Your workspaces in the cloud

No codespaces
You don't have any codespaces with this repository checked out

Create codespace on main

Learn more about codespaces...

Codespace usage for this repository is paid for by MSE-QualityAssurance

README.md

17 623 Quality Assurance: Recit

Installation

GitHub Codespaces

In the current repository, go to `<> Code > Codespaces > Create codespace on main`. Wait until the container finishes building.

Local Setup

Clone the current repository:

```
git clone git@github.com:MSE-QualityAssurance/recitation-2-f23.git
```

Run the following command for each of the projects:

About

17623 - Quality Assurance: Recitation 2 Fall 2023

Readme
Activity
0 stars
5 watching
0 forks
Report repository

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 2

pcanelas Paulo Canelas
nikki363

Languages

Java 100.0%