Exercises

Design Patterns: Observer

Java SE and Java EE patterns and best practices

João Miguel Pereira – http://jpereira.eu

0 Prerequisites, assumptions and notes

- Have Maven 2 installed in your computer
- Have Eclipse installed in your computer (Recommended: Indigo Version)
- I'm assuming you're running the exercises in Ubuntu
- It's recommended that you place all Design Pattern exercises under a common directory. For example:
  `${user.home}/javatraining/designpatterns`

  During the exercises I will refer this directory as
  `${designpatterns.exercises.folder}`

1. In every exercise I will refer the directory where you are working as `${project.dir}`.

1 Quick Start Exercise

You will put your hands on a small program and apply the Adapter Design Pattern.

1.1 Checkout code and create eclipse project

In this step you will checkout the code to `${project.dir}`.

Complete the following tasks.↓

1. Go to the `${project.dir}` directory

```
cd ${project.dir}
```

2. Checkout the code from code.google.com

```
svn checkout
http://javatrainings.googlecode.com/svn/designpat
terns/trunk/observer
```

3. Enter the created directory and run the tests to check that everything is ok.

```
mvn test
```

4. Enter the created folder and generate the eclipse project

```
mvn eclipse:eclipse
```

5. Import project into eclipse

✓ *you're done! You have now the project ready to refactor.*

1.2 Implement Observer pattern

Complete the following tasks.↓

1. Open the project **observer** with eclipse
2. Analyze package `eu.jpereira.trainings.designpatterns.behavioral.observer.observers`
3. Open test class `ApplianceEventObservableTest` and run the tests
4. Fix the tests:
    a. Open `EventMapper` and extend it from the interface `ApplianceEventObservable`
    b. Open `AbstractEventMapper` and implement the methods from interface `ApplianceEventObservable`
    c. Uncomment the commented code in `ApplianceEventObservableTest`
5. Run the tests

✓ *You're done.*