



Object-Oriented Design Lecture

Test Driven Development

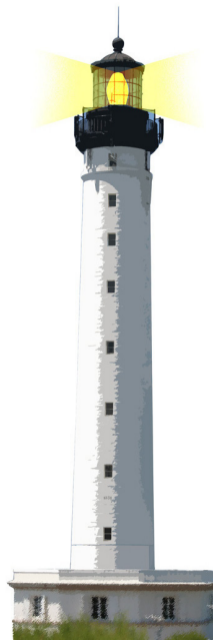
S. Ducasse

<http://stephane.ducasse.free.fr>

<http://car.mines-douai.fr/luc>



<http://www.pharo.org>



Outline

- Automated Tests
- Example of TDD
- Rethinking it



Automated Tests are your Life Insurance

- Our brain is too small to remember everything
- Our brain focuses on the last action
- You write it once and you run it million times
- Programming is modeling the world and the world is changing



Automated Tests

- Tests do not avoid breaking your system
- But they show what you broke



Automated Tests Let Software Evolve

- Tests make you **bold** in regards of changes
- Tests lower the fear of breaking
 - You can try and run the tests to get an idea
 - You can explore...



An Automated Test

In an automated test, we

- Create a context: Create an empty set
- Send a stimulus: Add twice the same element
- Check the results: Check that the set contains only one element



Set TestCase in Pharo

```
TestCase subclass: #SetTestCase
```

```
...
```

```
SetTestCase >> testAdd
```

```
| empty |
```

```
empty := Set new. "Context"
```

```
empty add: 5. "Stimulus"
```

```
empty add: 5.
```

```
self assert: empty size equals: 1 "Check"
```

```
SetTestCase run: #testAdd
```



Set TestCase in Java

```
class SetTestCase {  
    ...  
}
```

```
@Test  
public void testAdd(){  
    Set empty = new Set(); //Context  
    empty.add(5); //Stimulus  
    empty.add(5);  
    assertEquals(empty.size(), 1); "Check"  
}
```


TDD is about the flow

Write test first! Yes

"Whenever you are tempted to type something into a print statement or a debugger expression, write it as a test instead" - Martin Fowler



About the flow: Write your test

```
CounterTest >> testIncrement
| c |
c := Counter new.
c count: 10.
c increment; increment.
self assert: c count equals: 12
```

```
@Test
public void testIncrement(){
    Counter c = new Count();
    c.setCount(10);
    c.increment();
    c.increment();
    assertEquals(c.getCount(), 12);
}
```



About the flow: Run your test

- It is red!
- Else you increase your test number
- When you are lucky, use XTDD and enjoy coding in the debugger



About the flow: Make your test pass

- Work until your test get green!



About the flow: Rerun all the tests

- Now you are super smart but run all your tests!
- Fix the broken tests
 - Either a test is now wrong
 - Or you broke something for real
- Commit and take a break



Why writing test first

- You specify what you want to get
- You get a clear context
- You can debug on the spot



If we have to keep one practice: Tests

- You specify what you want to get
- You get a clear context
- You can debug on the spot



A course by

Stéphane Ducasse

<http://stephane.ducasse.free.fr>

and

Luc Fabresse

<http://car.mines-douai.fr/luc>



Except where otherwise noted, this work is licensed under CC BY-NC-ND 3.0 France

<https://creativecommons.org/licenses/by-nc-nd/3.0/fr/>