

# Tutorial 3

- Introduction to a unit testing tool, JUnit
- Introduction to a version control tool, RCS

## An Introduction to JUnit

- JUnit is a simple framework to write repeatable tests. It is an instance of the xUnit architecture for unit testing frameworks.
- Using *JUnit* you can cheaply and incrementally build a test suite that will help you measure your progress, spot unintended side effects, and focus your development efforts.

## Example: A Class Under Coding

```
class Money {
    private int fAmount;
    private String fCurrency;
    public Money(int amount, String currency) {
        fAmount= amount;
        fCurrency= currency;
    }
    public int amount() {
        return fAmount;
    }
    public String currency() {
        return fCurrency;
    }
}
```

M. Hu

Tutorial 3

3

## Add a New Method to the Class

- When you add two Moneys of the same currency, the resulting Money has as its amount the sum of the other two amounts.

```
public Money add(Money m) {
    return new
    Money(amount()+m.amount(),
    currency());
}
```

M. Hu

Tutorial 3

4

## How to Use JUnit to Write Test Case

- You implement a test in a subclass of TestCase. To test our Money implementation we therefore define MoneyTest as a subclass of TestCase. In Java, classes are contained in packages and we have to decide where to put MoneyTest. Our current practice is to put MoneyTest in the same package as the classes under test. In this way a test case has access to the package private methods. We add a test method testSimpleAdd, that will exercise the simple version of Money.add() above. A JUnit test method is an ordinary method without any parameters.

## The Test Case (Test Method)

```
public class MoneyTest extends TestCase {
    //...
    public void testSimpleAdd() {
        Money m12CHF= new Money(12, "CHF");
        // (1)
        Money m14CHF= new Money(14, "CHF");
        Money expected= new Money(26, "CHF");
        Money result= m12CHF.add(m14CHF);
        // (2)

        Assert.assertTrue(expected.equals(result));
        // (3)
    }
}
```

## Analysis of the Test Case

The testSimpleAdd() test case consists of:

1. Code which creates the objects we will interact with during the test. This testing context is commonly referred to as a test's *fixture*. All we need for the testSimpleAdd test are some Money objects.
2. Code which exercises the objects in the fixture.
3. Code which verifies the result.

## References

- For more information  
<http://junit.sourceforge.net/>
- To download Junit  
<http://junit.sourceforge.net/projects/junit/>

# An Introduction to RCS

- What is RCS
- How to use RCS
  - check in a file
  - check out a file
  - locking
- Reference

<http://www.gnu.org/software/rcs/rcs.html>