Integraion validation and unit tests

Prepare work environment

- upload Tests.zip in your workspace.
- Using it create a new projet Test-osssature.
- This project is configured using JUnit.
- You have to change the configuration path in order to include util.jar .

.

Executing existing tets

Open JUnit_DocumentTest and read existing test methods. Execute the unit tests of this class in Eclipse: *Run*, then *Run as* then *JUnit test* ;.

As an example of a test error, modify the method testReachableStates by removing d1.metEmpruntable(). Run JUnit_DocumentTest and observe this error.

Unit tests

Update the JUnit_DocumentTest.java to add the missing test. You have to add the right assertions for each test and specify if needed the exceptions managing the test scenario (for example, like the following annotation « @Test(expected=OperationImpossible.class) »). Bellow are the missing tests:

1. « borrow number» : borrow a document 5 times and check that the number of borrowed documents increases correctly;
2. « test 4.1 » : check that it is impossible (exception raised) de return a document that is not borrowed;
3. « double borrow »: check that is impossible to borrow a document that has been already borrowed
4. « exception of incorrect constructor » : check there is a raised exception when the input genre is null in the constructor of the class Document.

Integration tests

Being inspired by the integration tests of the class IntegrationTest, complete this class with the integration tests for the return document scenario « restituer » that check the interactions between FicheEmprunt, Document dans Client. The two cases that we propose to test are the following:

1. The borrowed item is returned on time :
   - Create a document,
   - Update it as borrowable,
   - Create a client,
   - Create a borrow file « fiche d'emprunt »
   - Return a document by calling a method of FicheEmprunt and check it from the document side (the document can be borrowed again) and client side (the borrowed items decreased),

2. the borrowed item has been returned late :
    - The scenario is similar to the previous test. Call Datutil.addAuJour(...) to progress artificially in the time, call verifier() of FicheEmprunt to test if the borrowed item is late, call premierRappel() to identify the client who is late.

Validation tests

The validation tests are in AcceptanceTest which contains the two first tests of the decision table presented in the course.

1. Write the validation test to borrow a document « emprunter un document » for at least one of the elements of the decision table (3-7). Start by the tests 4, 5 and 6, and finish with tests 7 and 3.
2. Check that if « genre » remove cannot be done if a document refers to this genre,
    - Write the corresponding validation test.