

Test d'integration

Test d'integration : test executable d'un composant, bouchons (Mock): pour ne pas executer toutes les dependances. On ne veut tester qu'un composant Composant de test heberger Test JUnit

3) et 4.

```
public class TestAbos {
    @Test
    public void testCategorie(){
        IAbos a = ComposantFactory.createAbos();
        assertEquals(a.getNbJmax("u007") , 30) // priv
        assertEquals(a.getNbJmax("u008") ,30) // abo
        assertEquals(a.getNbJmax("u009") , 15) // occasionel
    }
    /*
    Idem pour a.getNbEmpMax()
    */
    public void testInterdit(){
        assertFalse(a.isInterdit(u007))
        a.setInterdit(u007 , true)
        assertTrue(a.isInterdit(u007))
        a.setInterdit(u007 , false)
        assertFalse(a.isInterdit(u007))
    }
}

public class TestEmps{
    @Test
    IAbos a = ComposantFactory.createAboBouchon();
    IExs e = ComposantFactory.createEcBouchon();
    IEmps emps = ComposantFactory.createEmprunt();
    asserTrue(emps.demarrerEmprunt(u007))
    emps.ajouterEX("20kex1")
    asserFalse("20kex9")
}
```

Test Unitaire

7.

- Code exécutable qui valide un autre code : e.g. Junit
- Exécuté dans un cadre qui permet de détecter et d'analyser les erreurs Objectif : • Tester l'ensemble du code d'une classe, a priori en isolation

8.

les metrique :

- Etats : couverture des lignes de code (line coverage)

- Transitions : couverture des branchements (branch coverage)
- On voudrait que le jeu de test exerce toutes les lignes de l'implantation