Analyse de code

Effectuer des tests système en lançant l'application et en essayant de couvrir le maximum d'états possible. Dans le cas d'une défaillance rencontré, il est important de détailler.

1. Y a t-il une défaillance au niveau validité ?
2. Y a t-il une défaillance au niveau facilité d'utilisation ?
3. Y a t-il une défaillance au niveau performance ?
4. Y a t-il une défaillance au niveau fiabilité ?
5. Y a t-il une défaillance au niveau sécurité ?
6. Y a t-il une défaillance au niveau maintenabilité ?
7. Y a t-il une défaillance au niveau portabilité ?

Tests unitaires

8. Importer le source sous Eclipse.
9. Créer un dossier TEST au même niveau du dossier source, suivre également l'arborescence qui a été adopté au niveau du dossier source.
10. Créer pour chaque classe source, une classe test (ex, List.java ⇒ TestList.java)
11. Définir pour chaque méthode au minimum un cas de test. Il est important de diversifier les cas de test en utilisant les assertions suivantes :
    `assertEquals, assertFalse, assertTrue,`
    `assertSame, assertNotSame, assertNull,`
    `assertNotNull, assertArrayEquals, assertThat.`
12. Utiliser les annotations suivantes :
    `@Test, @Before, @After,`
    `@BeforeClass, @AfterClass,`
    `timeout, @Ignore`

Dans cette partie, vous allez être amener à produire un document (à déposer dans Moodle) et des classes de test des applications (à déposer aussi sur Moodle).
==========

Contact : Nadjib LAZAAR.