

ANNEXE 15 Atelier – Les Tests JUnit

Qu'est-ce que JUnit ? C'est une librairie de classes permettant de faire des tests unitaires pour le langage Java, créée par Kent Beck et Erich Gamma (www.junit.org)

Qu'est-ce qu'un test unitaire ? C'est un test qui permet de vérifier une partie / un module d'un programme. Dans le cadre de techniques comme l'Extreme Programming, l'utilisation de tests unitaires prend tout son sens. On suggère même de coder les classes de tests avant de coder l'application elle-même !

→ Jeu d'Échecs : nécessité de tester nos classes avant d'avoir l'interface graphique en main sinon la correction sera trop difficile en raison du nombre élevé de classes et de lignes de code → utiliser JUnit

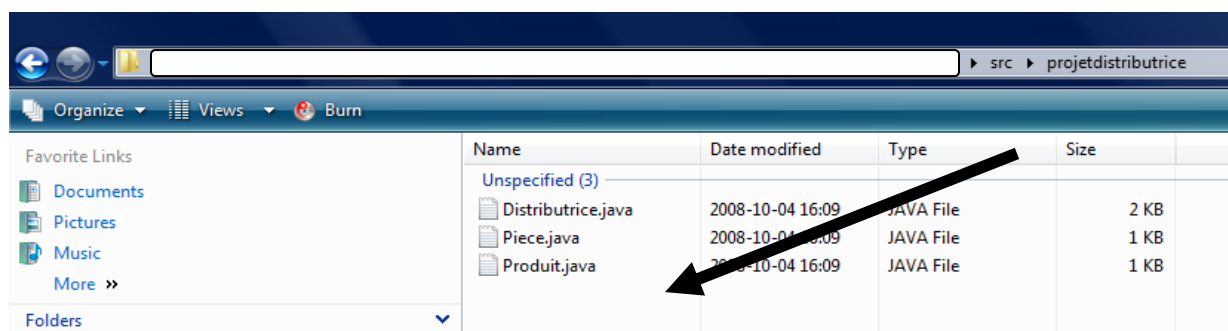
IMPORTANT

JUnit est une librairie de classes ne faisant pas partie de l'API Java de base. On doit donc ajouter la librairie à chacun des projets utilisant des tests JUnit.

Procédure

Exercice

1. Si ce n'est pas déjà fait, copier le dossier `ProjetDistributrice` présent dans `Donnees/B33` dans votre J.
2. Copiez le fichier `ModeleJUnit.java` (présent dans `Données/B33r`) dans le dossier du nom du package présent dans le dossier `src` de votre projet (où sont situés vos autres fichiers .java)



3. Ouvrir le projet dans JDeveloper

4. Adaptez tout de suite la classe de tests `ModeleJUnit` en y ajoutant le nom du package approprié au début de la classe et dans la méthode `main` de la classe.

5. Étudions ce projet, représentant une distributrice à friandises. Nous allons compléter la classe de tests. Comme son nom l'indique, vous pourrez utiliser `ModeleJUnit` comme modèle de classe de tests JUnit.

Types de méthodes de tests

A) Méthodes pré-test (annotation `@Before` devant l'entête de la méthode) : _____

B) Méthodes de test (annotation `@Test`) : _____

C) Méthodes après-test (annotation `@After`) : _____

6. Réalisez une méthode de test vérifiant la méthode `selectionCode` de la classe `Distributrice`

7. On peut également tester les méthodes qui ne retournent rien. Par exemple, faites un test vérifiant le comportement de la méthode `recevoirPiece` suite à l'ajout de deux `Pieces` dans la machine distributrice.

8. Tester la méthode `obtenirProduit`

NB

*** Les différentes méthodes de test sont appelées aléatoirement par le programme

***Différence entre `Failure` et `Error` : _____
