 JUnit et Eclipse

|  |  |  |
| --- | --- | --- |
|  |  |  |

JUnit est un framework open source pour réaliser des tests unitaires sur du code Java. Le principal intérêt est de s'assurer que le code répond toujours au besoin même après d'éventuelles modifications.

Le but est d'automatiser les tests. Ceux ci sont exprimés dans des classes sous la forme de cas de tests avec leurs résultats attendus. JUnit exécute ces tests et les comparent avec ces résultats.

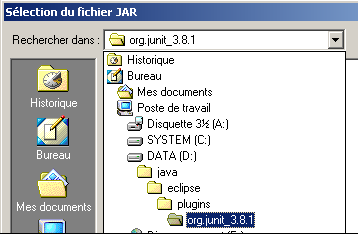
Avec Junit, l'unité de tests est une classe dédiée qui regroupe des cas de tests. Ces cas de tests exécutent les tâches suivantes :

* création d'une instance de la classe et de tout autre objet nécessaire aux tests
* appel de la méthode à tester avec les paramètres du cas de test
* comparaison du résultat obtenu avec le résultat attendu : en cas d'échec, une exception est levée

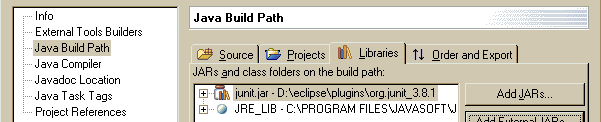
11.1. Paramétrage de l'environnement

Pour pouvoir utiliser JUnit dans un projet, il faut ajouter le fichier junit.jar dans le classpath du projet. Il faut pour cela :

* sélectionner les propriétés du projet
* dans l'onglet librairies, cliquer sur le bouton « Add External Jar »
* sélectionner le fichier junit.jar dans le répertoire plugins/org.junit\_3.8.1 du répertoire d'Eclipse



Eclipse ajoute le fichier junit.jar au classpath

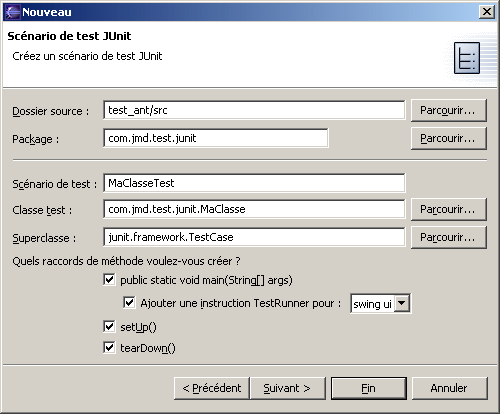


Dans ce chapitre, la classe suivante est utilisée comme classe à tester avec JUnit.

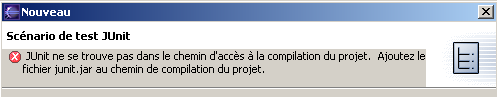
|  |
| --- |
| Exemple : |
| package com.moi.test.junit;  public class MaClasse {  public int additioner(int a, int b) {  return a + b;  }  } |

11.2. Ecriture des cas de tests

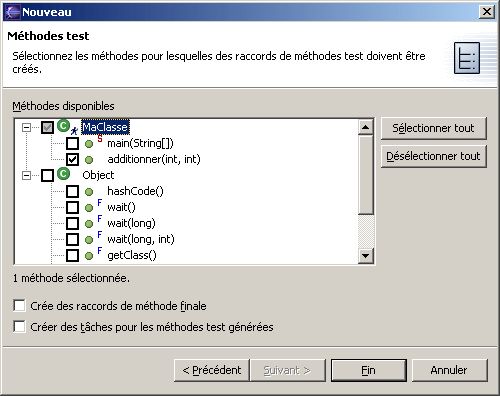
Pour utiliser JUnit, il faut créer une classe qui va contenir les cas de test. Il faut créer une nouvelle entité de type " Java / JUnit / Scénario de test ".



Si le fichier junit.jar n'est pas inclus dans le classpath du projet, un message d'erreur est affiché et il est impossible de poursuivre l'exécution de l'assistant.



Cliquez sur le bouton "Suivant".



Il faut compléter la classe générée selon les besoins : par exemple, ajouter un attribut qui va contenir une instance de la classe à tester, ajouter l'instanciation de cette classe dans la méthode setUp() et libérer cette instance dans la méthode tearDown().

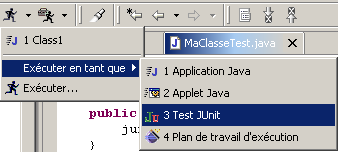
Il faut ajouter les traitements nécessaires dans les méthodes testXXX() en utilisant l'API de JUnit.

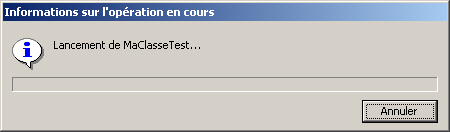
|  |
| --- |
| Exemple : |
| package com.moi.test.junit;  import junit.framework.TestCase;  public class MaClasseTest extends TestCase {     private MaClasse maClasse = null;     public MaClasseTest(String arg0) {           super(arg0);     }     public static void main(String[] args) {           junit.swingui.TestRunner.run(MaClasseTest.class);     }     protected void setUp() throws Exception {           super.setUp();           maClasse = new MaClasse();     }     protected void tearDown() throws Exception {           super.tearDown();           maClasse = null;     }     public void testAdditioner() {           assertTrue(maClasse.additioner(2,2) == 4);     }  } |

JUnit utilise l'instrospection pour exécuter les méthodes commençant par test.

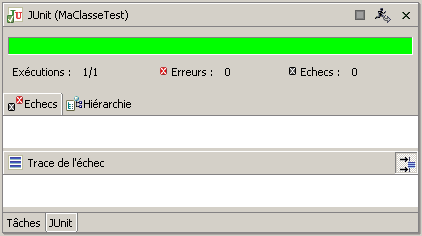
11.3. Exécution des cas de tests

Pour exécuter les tests, il faut exécuter la classe en tant que « Test JUnit  ».





Eclipse exécute les tests et affiche le résultat dans une vue dédiée.



Si tous les cas de tests ont été exécutés avec succès, une ligne verte est affichée.

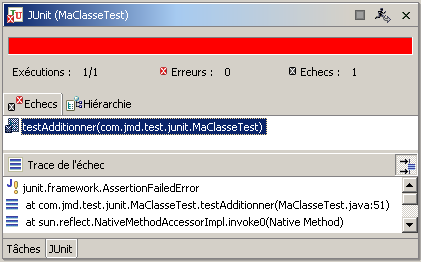
Cette vue contient deux onglets :

* « Failures » : contient la liste des cas de tests qui ont échoués
* « Hierarchy » : contient une arborescence des cas de tests

Dans le cas où un ou plusieurs tests échouent, la ligne est rouge.

|  |
| --- |
| Exemple : si le cas test suivant est ajouté : |
| public void testAdditioner() {              assertTrue(maClasse.additioner(2,2) == 4);              assertTrue(maClasse.additioner(2,3) == 4);        } |

Une exécution de l'exemple précédent permet d'avoir le résultat suivant :



11.4. Le support de JUnit 4

|  |  |
| --- | --- |
| http://www.jmdoudoux.fr/java/dejae/images/eclipse32.png | Eclipse 3.2 propose un support de la version 4 de JUnit. L'assistant de création d'un scénario de test permet de sélectionner la version de JUnit à utiliser.  http://www.jmdoudoux.fr/java/dejae/images/junit011.png  Pour ajouter la bibliothèque au classpath du projet, il suffit de cliquer sur le lien « Cliquez ici »  http://www.jmdoudoux.fr/java/dejae/images/junit012.png  Le fichier jar correspondant est ajouté et il suffit simplement d'appuyer sur le bouton « OK » pour valider l'ajout de la bibliothèque.  Le principe est identique lorsque c'est la version 4.0 de JUnit qui est sélectionnée.  http://www.jmdoudoux.fr/java/dejae/images/junit013.png  Cliquez sur le bouton « Suivant ».  http://www.jmdoudoux.fr/java/dejae/images/junit014.png  Sélectionnez les méthodes à tester dans le scénario et cliquez sur le bouton « Terminer »  http://www.jmdoudoux.fr/java/dejae/images/junit015.png  La classe générée prend en compte les fonctionnalités de la version 4 notamment les annotations.  L'assistant de code propose l'annotation @test.  http://www.jmdoudoux.fr/java/dejae/images/junit016.png  Le menu « Exécuter » propose automatiquement l'option « Exécuter en tant que / Test JUnit  ».  http://www.jmdoudoux.fr/java/dejae/images/junit017.png  En cas d'échec, les erreurs sont précisées dans la partie « Trace de l'échec ».  http://www.jmdoudoux.fr/java/dejae/images/junit018.png  L'assistant de code propose aussi l'annotation @Ignore pour ne pas prendre en compte une méthode de test.  http://www.jmdoudoux.fr/java/dejae/images/junit019.png  La vue JUnit propose un historique des tests effectués : pour afficher la liste des tests effectués, il suffit de cliquer sur le bouton http://www.jmdoudoux.fr/java/dejae/images/junit020.png.  http://www.jmdoudoux.fr/java/dejae/images/junit021.png  Il suffit de sélectionner le test à exécuter et de cliquer sur le bouton « OK ».  Il est possible de supprimer un test en le sélectionnant et en cliquant sur le bouton « Supprimer » ou de supprimer tous les tests en utilisant le bouton « Supprimer tout »  En cliquant sur la petite flèche du bouton, la liste apparaît directement dans le menu déroulant.  http://www.jmdoudoux.fr/java/dejae/images/junit022.png  L'option « Historique ... » permet d'ouvrir la boîte de dialogue « Exécutions de test ».  L'option « Effacer les éléments terminés » permet de vider l'historique. |