

Chapitre 11

Assertion en Java

Le mécanisme d'assertion permet au programmeur de vérifier dynamiquement des conditions. Ces vérifications sont optionnelles et généralement présentes lors du processus de développement et absentes lors de l'emploi final. Lorsque la condition n'est pas vérifiée, une exception est levée.

Exemple

```
public class AssertTest1 {  
  
    public static void main( String args[] ) {  
        assert args.length >= 2 && args.length <= 4;  
        System.out.println("->" + args[3] );  
    }  
}
```

Cet exemple suppose qu'il reçoit de 2 à 4 paramètres.

Note: Le mécanisme d'assertion existe depuis JDK-1.4.

La syntaxe du langage est modifiée pour supporter les nouvelles formes de JDK-1.4. Pour éviter des problèmes d'incompatibilité le programmeur doit activer la nouvelle syntaxe lors de la compilation de son application. De plus, le programmeur doit activer les assertions lors de l'exécution pour que les vérifications soient effectuées.

Compilation et exécution

```
% javac -source 1.4 AssertTest1.java  
  
% java AssertTest1  
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException  
at AssertTest1.main(AssertTest1.java:5)  
  
% java -ea AssertTest1  
Exception in thread "main" java.lang.AssertionError  
at AssertTest1.main(AssertTest1.java:4)
```

Le mécanisme d'assertion permet de donner une valeur textuelle à l'assertion avec la syntaxe:

```
assert expression1 : expression2;
```

Exemple

```
public class AssertTest2 {
    public static void main( String args[] ) {
        assert args.length >= 2 && args.length <= 4 : \
            "Invalid params (" + args.length + ")";
        System.out.println("->" + args[3] );
    }
}
```

Compilation et exécution

```
% java -ea AssertTest2
```

```
Exception in thread "main" java.lang.AssertionError: Invalid params (0)
at AssertTest2.main(AssertTest2.java:4)
```

Note: Il est possible d'activer ou de désactiver le mécanisme d'assertion via les fonctions du `ClassLoader`.