

## TP 02 : Les assertions JUnit

JUnit fournit des méthodes statiques pour tester certaines conditions via la classe **Assert**. Ces assertions (*assert statements*) commencent généralement avec *assert*. Ils vous permettent de spécifier le message d'erreur, le résultat réel d'exécution et le résultat attendu. Une assertion compare la valeur réelle renvoyée par un test à la valeur attendue. Elle génère une *AssertionException* si la comparaison échoue.

Le tableau suivant donne un aperçu de ces méthodes. Les paramètres entre crochets sont facultatifs et de type *String* :

Méthode	Rôle
<i>assertEquals([message,]Object a, Object b)</i>	Vérifie que les objets a et b sont égaux
<i>assertSame([message,]Object a, Object b)</i>	Vérifie que a et b sont des références vers le même objet
<i>assertNotSame([message,]Object a, Object b)</i>	Vérifie que a et b ne sont pas des références vers le même objet
<i>assertNull([message,]Object o)</i>	Vérifie que l'objet o est null
<i>assertNotNull([message,]Object o)</i>	Vérifie que l'objet o n'est pas null
<i>assertTrue([message,]boolean e)</i>	Vérifie que l'expression e est vraie
<i>assertFalse([message,]boolean e)</i>	Vérifie que l'expression e est fausse
<i>assertArrayEquals([message,] expected, actual)</i>	Vérifie que deux objets de type Array sont égaux ou non
<i>fail([message,])</i>	Provoque l'échec du test

TABLE 1 – Méthodes d'assertions en JUnit

## 1 Exercice

Écrivez la classe **MethodsUnderTestTest** qui décrit les tests unitaires des méthodes de la classe **MethodsUnderTest**. En particulier, pour chacun des tests, précisez en commentaire l'objectif du test ainsi que le résultat attendu. Pensez à tester aussi bien les cas qui doivent réussir que les cas qui doivent lever une exception : l'objectif est de couvrir un maximum de cas. Pensez également aux cas aux limites.

```
public class Livre {  
    private String titre, auteur;  
    private int nbPages;  
}
```



```

    private double prix;

    public Livre(String titre, String auteur, int nbPages, double prix){this
        .titre=titre;
        this.auteur = auteur; this
        .nbPages = nbPages; this
        .prix=prix;
    }
}

```

Livre.java

```

public class MethodsUnderTest {

    private Livre l = null;

    public int add(int a, int b){
        int res = a;
        if (b > 0) {
            while (b-- != 0) {
                res++;
            }
        } else if (b < 0) {
            while (b++ != 0) {
                res--;
            }
        }
        return res;
    }

    public int mult(int a, int b){
        int res = 0;
        while (b > 0) {
            if (b % 2 != 0) {
                res = res + a;
                b = b - 1;
            }
            a = 2 * a;
            b = b / 2;
        }
        return res;
    }

    public double calculatePI(int n)
    { double PI, s = 0;
      int i;
      if (n < 0) {
          PI = 0;
      }
      for (i = 0; i < n + 1; i++) {
          s = s + ((java.lang.Math.pow(-1, i)) / ((2 * i) + 1));
      }
      PI = 4 * s;
    }
}

```

32

34

36

38

40

42

```

44     return (PI)
    }
    ;

46 // Accès autorisés s'il existe une suite de 4      chiffrés consécutifs dans
// suitesaisie      res      la
public boolean verifCode (int T[], int n) { int
48     res = 0;
    int i = 1;
50     while (res < 3 && i < n) {
        if (T[i] == T[i-1] + 1) {
            res = res + 1
52             i = i + 1
            ;
        } else
54             { res
                = 0;
                i = i + 1
56             }
        return (res == 3);
    }
}

58 // la Rotation à gauche d'un tableau
public int[] leftRotating (int T[], int n)
60     { int T2[] = new int[n];
        for (int i = 0; i < n; i++) { T2[ (
62             (i + 4) % 5 ) ] = T[i];
        }
        return (T2);
64     }
}

66 public Livre add Livre (String titre, String auteur, int nbpages, double prix)
    {
        if (!titre.equals("") && !auteur.equals(""))
68             { l = new Livre (titre, auteur, nbpages, prix);
            }
        return (l);
70     }
}

72 public Livre get Livre () {
    return l;
    }
}
74

```