

## TP2 : spécification, implémentation et validation "Copie de Tableau"

### Installation

1. Ouvrez une "fenêtre de commandes" Windows (onglet Démarrer, puis Exécuter "cmd"). Dans la fenêtre de commandes, tapez la commande suivante ou faites un copier/coller:  

```
> set Path=%Path%;\\info\Bibliotheque\MF-dev\TP-EscJava\windows\escjava\bin
```
2. Placez-vous dans le répertoire <REPERTOIRE\_TP>\Exercices\1.Tableaux  

```
> Z : > CD <REPERTOIRE_TP>\Exercices\1.Tableaux
```

Pour utiliser escjava, tapez la commande (Windows) suivante:

```
> escjava -loopSafe CopieDeTableaux.java
```

Le répertoire \\info\Bibliotheque\MF-dev\TP-EscJava\Doc contient de la documentation sur ESC/Java.

### Expliquez les directives suivantes :

```
invariant
loop_invariant
requires
ensures
assert
```

### Méthode affecter

1. Dans la spécification de la méthode affecter, trouvez la bonne valeur pour les ??? . Une fois les points d'interrogation remplacés, retirez les premiers commentaires de la seconde ligne.  

```
/*@ requires t != null;
// //@ ensures (\forall int j; j>=0 & j < t.length ==> t[j] == ???);
```
2. Dans le code de la méthode affecter, complétez la seconde directive loop\_invariant (remplacez les "???" par l'expression correcte). Une fois les points d'interrogation remplacés, retirez les premiers commentaires de la seconde ligne.  

```
/*@ loop_invariant i>=0;
// //@ loop_invariant (\forall int j; j>=0 & ??? ==> t[j] == x);
```
3. Validez votre code (aucune mise en garde ne doit être générée par escjava).

### Méthode copie

La méthode copie copie le contenu de t1 dans t2

4. Rédigez la spécification formelle (directives ensures et requires) de la méthode copie.
5. Codez la méthode copie. Le code doit être conforme à cette spécification. (esc/java doit le vérifier!). Attention, les assertions de la méthode test doivent également être vérifiées.

```

public class CopieDeTableaux {

    public CopieDeTableaux() {}

    //@ requires t != null;
    // //@ ensures (\forall int j; j>=0 & j < t.length ==> t[j] == ???);
    public static void affecter(int[] t, int x){
        int i = 0;
        //@ loop_invariant i>=0;
        // //@ loop_invariant (\forall int j; j>=0 & ??? ==> t[j] == x);
        while( i < t.length){
            t[i] = x;
            i++;
        }
    }

    public static void copie(int[] t1, int[] t2) { }

    public void test() {
        int[] t1 = {1,9,11,15,56,78,89};
        //@ assert t1[5] == 78;
        int[] t2 = new int[t1.length];
        copie(t1,t2);
        //@ assert t2[5] == 78;
    }
}

```