

**TP4 R201 S2**  
**B. Lemaire et H. Azzag**

**A. Définir et utiliser des tableaux**

Considérons le code source incomplet suivant :

```
Public class TableauBooleens {
    boolean[] tab;

    public TableauBooleens (boolean[] t) {...}
    public TableauBooleens (TableauBooleens tb) {...}
    public boolean get(int i) {...}
    public void set(int i, boolean val) {...}
    public int taille() {...}
    public TableauBooleens et(TableauBooleens tb) {...}
    public boolean et() {...}
    public String toString() {...}

    /*****/

    public static void main(String[] args) {
        boolean[] t1 = {false, false, false, false};
        TableauBooleens a=new TableauBooleens(t1);
        System.out.println("a avant : " + a);
        t1[0]=true;
        System.out.println("a après : " + a);
        TableauBooleens b=new TableauBooleens(a);
        t1[1]=true;
        b.set(2, true);
        System.out.println("a : " + a);
        System.out.println("b : " + b);
    }
}
```

1. Écrire le constructeur `TableauBooleens (boolean[] t)`. Ce constructeur ne doit pas créer de nouveau tableau.

2. Écrire la méthode `taille()` qui retourne le nombre d'éléments du tableau.
3. Écrire la méthode `toString()` qui représente un `TableauBooleens` sous la forme `|true|false|true|false|`, par exemple.
4. Pourquoi les `println()` des instructions 16 à 18 doivent afficher ce qui suit ?

a avant : `|false|false|false|false|`

a après : `|true|false|false|false|`

5. Écrire le second constructeur. Celui-ci doit créer une copie du tableau en paramètre.
6. Écrire les méthodes `get*(.)` et `set*(.)`.
7. Qu'affichent les instructions des lignes 22 à 23 ?
8. Écrire les deux méthodes `et(.)`
  - a. La première retourne un nouveau `TableauBooleens` dont chaque booléen est obtenu par un *et* logique entre les éléments de même rang de l'objet courant et de celui en paramètre.
  - b. La seconde réalise un *et* logique entre tous les éléments du `TableauBooleens` courant.
9. Compléter la méthode `main()` pour tester les deux méthodes précédentes sur des exemples pertinents.

## B. Définition de la classe *Triangle*

- A. Réécrire la classe `Triangle` (vu en cours) en mettant en variable d'instance un tableau de 3 points. Vous définirez les getter et setter.
- B. Implémenter la méthode `void translate(int dx, int dy)`