

TD 6

M.Adam – N.Delomez – JF.Kamp – L.Naert

10 août 2022

Objectifs du TD

- Apprendre à construire des boucles imbriquées

Exercice 1 (*)

Expliquer le fonctionnement et le rôle du programme suivant :

```
/**
 * Rôle à deviner
 * @author M.Adam
 */
class Saisie {
    final int LG_TAB = 10;

    void principal () {
        int[] valeurs = new int[LG_TAB];
        int i = 0;
        while (i != LG_TAB) {
            valeurs[i] = SimpleInput.getInt ("Entrer un entier : ");
            boolean trouve = false;
            int j = 0;
            while (!(trouve || j == i)) {
                if (valeurs[i] == valeurs[j]) {
                    trouve = true;
                }
                j = j + 1;
            }
            if (!trouve) {
                i = i + 1;
            }
        }
        System.out.println ("Toutes les valeurs sont saisies.");
        for (i = 0; i <valeurs.length; i = i + 1) {
            System.out.print (valeurs[i] + "\t");
        }
    }
}
```

Exercice 2 (**)

La méthode `saisirEtTrier()` a pour objectif d'insérer la valeur saisie dans le tableau. Les valeurs doivent être insérées de manière à ce que le tableau soit trié dans l'ordre croissant.

Compléter la boucle interne du programme. Évidemment, la méthode de construction des boucles doit être utilisée.

```
/**
 * Saisie de 10 valeurs dans un tableau
 * @author M.Adam
 */
class SaisieOrdre {
    final int LG_TAB = 10;
    void principal () {
        int[] valeurs;

        valeurs = saisirEtTrier ();

        System.out.println ("Toutes les valeurs sont saisies.");
        for (int i = 0; i < valeurs.length; i = i + 1) {
            System.out.print (valeurs[i] + "\t ");
        }
    }

    /**
     * Crée et saisit un tableau trié de LG_TAB entiers
     * @return tableau trié de LG-TAB entiers
     */
    int [] saisirEtTrier () {
        int[] t = new int[LG_TAB];
        int i = 0;

        while (i < t.length) {
            t[i] = SimpleInput.getInt ("Entrer un entier");
            // insertion de la valeur en ordre croissant dans t
            i = i + 1;
        }
        return t;
    }
}
```

Algotouch : <https://tinyurl.com/TD06AGT02>

Exercice 3 (***)

La méthode `intersection()` détermine les valeurs communes de deux tableaux :

```
/**
 * calcule les valeurs communes de deux tableaux
 * @param tab1    tableau d'entiers
 * @param tab2    tableau d'entiers
 * @return tableau d'entiers contenant les valeurs communes aux deux premiers tableaux
 */
int[] intersection (int [] tab1, int[] tab2)
```

Les tableaux ne contiennent pas de valeurs en double.

Écrire le code de la méthode en utilisant la méthode vue en cours.

