

TP 5

L. Naert – T. Ferragut – E. Lemonnier

9 octobre 2024

Objectifs du TP

- Construire des boucles imbriquées
- Faire des fonctions avec des passages par référence.

Exercice 1 : boucles imbriquées (**)

La méthode `sontTousDiff()` vérifie si les deux tableaux n'ont aucun entier en commun. La méthode rend vrai si les 2 tableaux n'ont aucune valeur commune et rend faux s'il existe au moins une valeur présente dans les deux tableaux.

```
/**
 * vérifie si deux tableaux n'ont aucune valeur commune
 * @param tab1 premier tableau
 * @param tab2 deuxième tableau
 * @return vrai si les tableaux tab1 et tab2 n'ont aucune valeur commune, faux sinon
 */
boolean sontTousDiff (int[] tab1, int[] tab2)
```

Écrire le code de la méthode `sontTousDiff()`.

Rendre le code de la méthode, le code des méthodes de test et l'exécution.

Exercice 2 : passage par référence - CT 2022 (**)

1. Écrire la méthode :

```
/**
 * décale les entiers d'un tableau d'une position vers la gauche
 * L'élément en 0 se retrouve à la fin du tableau
 * @param tab tableau d'entiers
 */
void decalerGauche (int[] tab)
```

Exemple : le tableau {3, 10, 6, 20, 7} devient {10, 6, 20, 7, 3}.

2. En utilisant `decalerGauche()`, écrire la méthode :

```
/**
 * décale les entiers d'un tableau de n positions vers la gauche
 * @param tab tableau d'entiers
 * @param n entier nombre de cases à décaler
 */
void decalerGaucheN (int[] tab, int n)
```

Exemple : le tableau `tab` contenant {3, 10, 6, 20, 7}, devient {20, 7, 3, 10, 6} après l'appel `decalerGaucheN(tab, 3)`;

3. Écrire la méthode :

```
/**
 * cherche l'indice de la premiere occurrence d'une valeur dans un tableau
 * @param tab tableau d'entiers
 * @param v valeur à chercher
 * @return l'indice de la première valeur v dans tab si v est dans tab, -1 sinon
 */
int indiceTab (int[] tab, int v)
```

4. Écrire la méthode :

```
/**
 * décale les valeurs d'un tableau de manière à ramener la valeur cherchée
 * à l'indice 0
 * Si la valeur n'est pas présente, le tableau n'est pas modifié
 * @param tab tableau d'entiers
 * @param v valeur à chercher
 */
void decaleValeur (int[] tab, int v)
```

Exemple : le tableau `tab` contenant {3, 10, 6, 20, 7} devient {20, 7, 3, 10, 6} après l'appel `decaleValeur(tab, 20)`.

Rendre le code complet (principal + méthodes) et des exemples d'exécution.

Exercice 3 (***)

La méthode `compteDiffVal()` compte le nombre de valeurs différentes dans un tableau passé en paramètre.

```
/**
 * compte le nombre de valeurs différentes dans un tableau
 * @param tab tableau d'entiers
 * @return le nombre de valeurs différentes du tableau
```

```
*/  
int compteDiffVal(int[] tab)
```

Par exemple, `compteDiffVal ({0, 0, 2, 3, 0, 2, 1, 3, 3, 0})` rend la valeur 4, car il existe 4 valeurs distinctes.

Écrire le code de la méthode `compteDiffVal()`.

Rendre le code de la méthode, le code des méthodes de test et le résultat d'exécution.

Exercice 4 (***)

Écrire une méthode `estSousChaine()` qui détermine si une chaîne est présente dans une autre.

```
/**  
 * teste si une chaîne est une sous-chaîne d'une autre  
 * @param mot chaîne de caractères  
 * @param phrase chaîne de caractères  
 * @return vrai ssi mot est présent dans phrase  
 */  
boolean estSousChaine (String mot, String phrase)
```

La comparaison s'effectue caractère par caractère.

Dans les tests, répondre (entre autres) aux questions :

- la chaîne `ses` est-elle dans `abcdsesdef` ?
- la chaîne `ses` est-elle dans `abcdef` ?
- la chaîne `ses` est-elle dans `abcdefse` ?

Rendre le code de la méthode, les méthodes de test et le résultat de l'exécution.

Exercice 5 (**)

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

Compléter la boucle interne du programme.

```
/**  
 * 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;  
}
```

Rendre le code de la méthode et des exemples d'exécution.