

IAD : TP9

BUT/INFO

version 2021-2022

Table des matières

- 1. Environnement de travail
- 2. Fonctions classiques de manipulation sur les tableaux en Java
- 3. Exercice : Sous-programme inverserTableau
- 4. Exercice : Programme eliminerDoublons
- 5. Exercice : Programme indicesMax
- 6. Pour aller plus loin ...
 - 6.1. Optimiser eliminerDoublons
 - 6.2. Optimiser indicesMax
- 7. Avant de partir

1. Environnement de travail

Tout comme dans les TPs précédents, réaliser les actions suivantes pour configurer votre environnement de travail.

1. Lancer BlueJ.
2. Créer et nommer (nous vous suggérons TP9_NumeroDeGroupe_Nom_Prenom) un nouveau projet où vous souhaitez placer vos sources java.

2. Fonctions classiques de manipulation sur les tableaux en Java

Nous nous apprêtons à concevoir des algorithmes sur des tableaux. Pour (1) rendre cette tâche plus aisée, (2) faire un petit rappel de certains fonctions utiles déjà vues, (3) progresser graduellement, nous proposons tout d’abord d’implémenter des sous-programmes d’aide à la manipulation de tableaux.

Créer une nouvelle classe Java `AlgosTableaux`, et y placer le code suivant :

Exemple 1. `AlgosTableaux.java`

```
import java.util.Scanner;

public class AlgosTableaux {

    /**
     * Demande à l'utilisateur de saisir un entier,
     * jusqu'à ce que
     * l'entier saisi soit entre les deux bornes en
     * paramètres.
     *
     * @param pfBorneInf IN : borne inférieure
     * @param pfBorneSup IN : borne supérieure
     *
     * @return un entier entre pfBorneInf et
     * pfBorneSup, compris
     */
    public static int saisieIntC (int pfBorneInf,
    int pfBorneSup){
        int valeur;
        Scanner clavier = new Scanner(System.in) ;
        System.out.println("Donnez une valeur
    comprise entre "+pfBorneInf+" et "+pfBorneSup+ "?");
        valeur = clavier.nextInt();
        while (valeur<pfBorneInf ||
    valeur>pfBorneSup){
            System.out.println("Erreur ! Donnez une
    valeur comprise entre "+pfBorneInf+" et
    "+pfBorneSup+ "?");
            valeur = clavier.nextInt();
        }
        return valeur;
    }

    /**
     * Demande à l'utilisateur :
     * 1. de saisir le nombre de cases qu'il
     * souhaite remplir, et
     * répète l'opération, jusqu'à ce que ce
     * nombre soit
     * acceptable
     * 2. de remplir les cases une à une.
     *
     * @param pfTab OUT : tableau à remplir
     *
     * @return le nombre de cases remplies dans le
     * tableau
     */
    public static int saisirTableau (int[] pfTab){
        int nbEl = 0 ;
        System.out.println("J'avais " + pfTab.length
            + " cases disponibles, et
    j'en ai rempli "
            + nbEl);

        return nbEl ;
    }

    /**
     * Affiche le tableau en paramètre.
     *
     * @param pfTab IN : tableau
     * @param pfNbEl IN : nombre de cases remplies
     * dans le tableau
     *
     */
    public static void afficherTableau(int[] pfTab,
    int pfNbEl) {
        System.out.println("Je n'ai pas encore
    implémenté la méthode afficherTableau !");
    }

    public static void principale() {

        /* Declaration des variables */
        int nbVal ; //nombre de valeurs a traiter
        int tab[] ; // tableau permettant de
        stocker les valeurs

        tab = new int[100] ;

        // Ajouter les appels aux algorithmes :
        // - à saisirTableau
        // - puis à afficherTableau
        // - puis ...

    }

}

/*****
Jeu d'essais pour AlgosTableaux
*****/
A vous !!!

*****/
```


Vous l’aurez compris, il est demandé de compléter les fonctions `saisirTableau` et `afficherTableau`, dont la javadoc doit vous permettre de comprendre les spécifications (ce qui est attendu).
Comme par complétude la fonction `principale` par des appels aux deux fonctions précédemment citées ; c’est cette fonction `principale` que vous exécuterez dans BlueJ lorsque vous testerez vos autres sous-programmes.

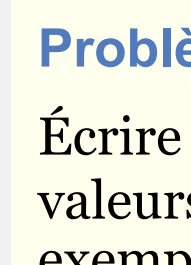
3. Exercice : Sous-programme inverserTableau

Problème

Ajouter dans le programme `AlgosTableaux` un sous-programme `inverserTableau` qui, à partir d’un ensemble de valeurs entières saisies dans un tableau inverse le contenu de ce tableau. Ainsi, le tableau {34, 1, 5, 6, 1, 7} deviendra {7, 1, 6, 5, 1, 34}. Il est **imposé** d’écrire la javadoc et la signature de la fonction avant toute chose.

Complétez le programme principal pour tester votre sous-programme en inversant les valeurs du tableau qui auront été saisies et affichées.

 Vous ne devez pas utiliser de tableau supplémentaire !

 Pour inverser le contenu d’un tableau, il peut être utile de gérer un indice de début et un indice de fin

4. Exercice : Programme eliminerDoublons

Problème

Écrire un programme `eliminerDoublons` qui, à partir d’un tableau de valeurs entières, construit et affiche cet ensemble sans doublon. Par exemple, à partir du tableau {0, -22, 10, -22, -22, 12, 10, 30, 10}, doit s’afficher : 0 -22 10 12 30. Avant toute chose, écrire la javadoc, la signature de la fonction et son appel dans la fonction `principale`.

-  Dans un premier temps, vous pouvez utiliser un second tableau pour stocker l’ensemble des valeurs sans doublons.
1. Vous donnerez le jeu d’essais en commentaires de votre programme
 2. Faire l’algorithme sur papier puis le codage.

5. Exercice : Programme indicesMax

Problème

Écrire un programme `indicesMax` qui identifie les indices des cases d’un tableau où se trouvent la plus grande valeur du tableau.

La documentation et la signature de la fonction sont les suivantes :

```
/*
 * Stocke les indices des occurences du maximum
 *
 * @param pfTab IN : un tableau de valeurs
entières
 * @param pfNbEl IN : le nombre de valeurs
 * @param pfIndice OUT : le tableau des indices
 *
 * @return le nombre d'occurrences du max
 */
public static int indicesMax(int[] pfTab, int
pfNbEl, int[] pfIndices) ;
```

Ainsi, ce bout de code :

```
int[] indices = new int[10] ;
int[] tab = {1,15,-2,15,6};
int nbOccurrences ;
nbOccurrences = indicesMax(tab, tab.length, indices);
afficherTab(indices, nbOccurrences); // AFFICHE: 1
3
```

doit afficher : 1 3.

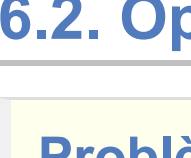
Faire un jeu d’essai, puis inclure et coder la fonction `indicesMax`.

6. Pour aller plus loin ...

6.1. Optimiser eliminerDoublons

Problème

Écrire un programme `eliminerDoublonsbis` qui, à partir d’un tableau de valeurs entières, modifie le tableau afin d’en éliminer les doublons et affiche ce tableau sans doublon.

-  Vous utiliserez **un seul tableau** pour saisir l’ensemble des valeurs à traiter et pour stocker l’ensemble des valeurs sans doublons.

6.2. Optimiser indicesMax

Problème

Si ce n’est pas déjà fait, écrire une fonction `indicesMaxbis`, qui partage les mêmes objectifs que `indicesMax`, mais qui les atteint en ne visitant chaque case du tableau qu’une seule fois.

7. Avant de partir

1. Enregistrer vos programmes sur **webetud**
2. N’oubliez-pas de vous déconnecter