

TP petites fonctions sur tableaux

Algorithmique – Programmation FIP (ING39)

V. Aponte, P. Courtieu

Le but de ce TP est de travailler EN BINÔMES (testeur + programmeur), l'écriture de contrats, des tests et le code de petites fonctions sur tableaux en java.

Exercice 1

Vous allez constituer un groupe de travail de 2 personnes. L'un de vous deux va jouer le rôle du *testeur* pendant un certain temps, l'autre du *développeur*. Il est recommandé de changer les rôles au moins une fois pendant la durée du Tp. Au départ, **le plus expérimenté en java parmi vous**, prendra le rôle du *testeur* et accessoirement, pourra aider le *programmeur* s'il a des difficultés. Considérez le squelette de programme suivant:

```
public class ArrayExamples {

    // Determine si deux tableaux sont égaux composante par composante.
    public static boolean eq(int [] arr1, int[] arr2) {
        return false;
    }

    // Retourne le tableau inverse de arr.
    public static int[] reverse(int [] arr) {
        return null;
    }

    // Recopie le tableau a entre les positions m jusqu'à n
    public static int[] sub(int[] a, int m, int n) {
        return null;
    }

    // Concatène deux tableaux
    public static int[] concat (int[] arr1, int[] arr2) {
        return null;
    }

    // Entrelace deux tableaux.
    public static int[] interleave (int[] arr1, int[] arr2) {
        return null;
    }

    // Calcule la valeur maximale d'un tableau ayant au moins une composante.
    public static int max(int[] arr) {
        return -1;
    }

    // Calcule l'indice de la valeur maximale d'un tableau ayant au moins une composante.
```

```

public static int maxIndex(int[] arr) {
    return -1;
}

// Teste si un tableau à deux dimensions est rectangulaire
// (pourquoi ce n'est pas nécessairement le cas en Java?)
public static boolean rect (int [][] a) {
    return false;
}

// Calcule la moyenne des composantes d'un tableau à deux dimensions.
public static int avg (int [][] a) {
    return 0;
}
}

```

1. Les commentaires qui décrivent ces méthodes sont des contrats incomplets et plutôt vagues: que se passe-t-il si un paramètre est null, vide, si une position n'est pas un indice valide?. De plus, ces commentaires ne sont pas au format javadoc. A deux, rédigez des contrats précis pour TOUTES ces méthodes au format javadoc. N'écrivez rien d'autre avant de finir de rédiger tous les contrats, committez et poussez.
2. L'un de vous deux (**et seulement l'un de vous**) va créer sur gitlab un projet COMMUN de nom Tp4Bis et va donner les droits d'accès à l'autre. Pendant toute la durée du tp vous allez travailler sur ce projet commun.
3. **Le plus expérimenté en java parmi vous**, prendra le rôle du *testeur* et va donc écrire les tests pour les 3 premières méthodes. L'autre prendra le rôle du développeur et va donc programmer ces 3 méthodes.
4. Régulièrement chacun de vous va commiter et pousser son travail.
5. Dès que possible, le *développeur* va utiliser les tests sur la forge pour tester chacune des méthodes qu'il doit écrire.
6. Le *développeur* ne passera pas à la prochaine méthode à écrire tant que tous les tests sur la forge n'ont pas réussi.
7. Procédez de cette manière pour le reste des méthodes. Il est recommandé de changer de rôle au moins une fois.