

Check-list n°4.1 : Opérations sur des matrices d'entiers

Jérôme Palaysi

Addionner.java

```
public class Additionner{

    /**
     * Donnée : deux matrices d'entiers P et Q de même dimensions
     * Résultat : retourne une nouvelle matrice égale à P+Q
     */
    public static int[] [] additionner(int P[] [], int[] [] Q){
        ...
    }
}
```

SommeDesDiagonales.java

```
public class SommeDesDiagonales{

    /**
     * Donnée : une matrice carrée M d'entiers
     * Résultat : retourne la somme des entiers des diagonales de M
     */
    public static int sommeDesDiagonales(int[] [] M){
        ...
    }
}
```

NombreDeZeros.java

```
public class NombreDeZeros{
```

```

    /**
     * Donnée : ...
     * Résultat : ...
     */
    public static int nombreDeZeros(int[] [] M){
        ...
    }
}

```

LigneAvecUnMaxDeZeros.java

Énoncé : Savoir sur quelle ligne d'une matrice donnée on trouve le plus de valeurs nulles.

```

public class LigneAvecUnMaxDeZeros{

    /**
     * Donnée : une matrice M de m>=1 lignes
     * Résultat : retourne le numéro 1<=i<=m d'une des lignes de M comportant
     *             le plus de 0
     */
    public static int ligneAvecUnMaxDeZeros(int[] [] M){
        ...
    }
}

```

IlExisteUneLigneAvecPlusieursZeros.java

```

public class IlExisteUneLigneAvecPlusieursZeros{

    /**
     * Donnée : ...
     * Résultat : ...
     */
    public static boolean ilExisteUneLigneAvecPlusieursZeros(int[] [] M){
        ...
    }
}

```

SontÉgales.java

L'opérateur `==` de Java compare les adresses des tableaux dans la mémoire, mais pas leur contenu. Par conséquent, `A==B` est faux, même si on a défini A et B de la façon suivante : `int[] A={1,2,3}; int[] B={1,2,3};`. La méthode à programmer ci-dessous¹ doit permettre, elle, de comparer le contenu des matrices.

```
public static bool sontÉgales (int[] [] P, int[] [] Q){  
    ...  
}
```

¹Sans utiliser de méthodes préexistantes dans Java comme `deepequals`.