

TD 3

Le type abstrait List<E>

1 Utilisation simple d'une liste

1. Filtrage : supprimer les multiples d'un entier dans une liste ;
2. Écrire une fonction prenant une liste en paramètre et retournant une nouvelle liste contenant tous les éléments de la liste paramètre mais sans doublon.
3. Écrire une fonction qui insère un *entier* à sa place dans une liste *d'entiers* triée. Si cet entier est déjà présent dans la liste, il ne sera pas inséré.

2 Grands nombres

On s'intéresse ici à la représentation de nombres entiers naturels très grands (plusieurs milliers de chiffres), dont la taille excède la capacité des types intégrés de Java.

Un entier naturel est représenté par la *liste de ses chiffres classés par exposant décroissant* ; par exemple, l'entier naturel 10 050 100 301 102 050 870 sera représenté par la liste (1, 0, 0, 5, 0, 1, 0, 0, 3, 0, 1, 1, 0, 2, 0, 5, 0, 8, 7, 0) ;

1. Écrire une fonction qui permet de savoir si un entier naturel représenté par la liste de ses chiffres est *divisible par trois*.
2. Écrire une fonction qui détermine si deux entiers naturels représentés par la liste de leurs chiffres sont égaux.
3. Écrire une fonction qui calcule la somme de deux entiers naturels représentés par la liste de leurs chiffres ; le résultat est une nouvelle liste.

3 Type abstrait représenté avec une liste

1. Un médecin souhaite s'équiper d'un système d'aide au diagnostic des maladies. Pour ce faire, il saisit un certain nombre de maladies et leur associe un ensemble de symptômes. Un symptôme est représenté par un *entier*, un ensemble de symptômes par une *liste triée d'entiers*. Une maladie est représentée par le type abstrait suivant :

```

1  public interface Maladie {
    /**
     * Constructeur
     * @param nom : nom de la maladie
     * @param symptomes : ensemble des symptômes associés à la maladie
6   */
    // public Maladie(String nom, List<Integer> symptomes);

    /**
     * getNom
     * @return nom de la maladie
11  */
    public String getNom();

    /**
     * getSymptomes : renvoie l'ensemble des symptômes associés à la maladie
     * @return l'ensemble des symptômes associés à la maladie
16  */
    public List<Integer> getSymptomes();
    }

```

- (a) Écrire une fonction de ressemblance entre deux ensembles de symptômes. Cette fonction renvoie le nombre d'éléments communs entre les deux ensembles. Elle sera utile pour comparer des symptômes remarqués avec les symptômes associés aux maladies connues.
- (b) Écrire une fonction prenant en paramètre une liste de Maladies (la base de connaissances du médecin) ainsi qu'un ensemble de symptômes (observés) et renvoyant une liste de maladies maximisant la ressemblance avec les symptômes.
À l'aide de la liste renvoyée par cette fonction, le médecin possèdera une première base de diagnostic.