

# Exercice R1

Le but de cet exercice est de gérer une liste d'entiers dont chacun représente une note de devoir entre 1 et 60.

| Marks |                                 |
|-------|---------------------------------|
| —     | alMarks : ArrayList<Integer>    |
| +     | addMark(pMark : int) : void     |
| +     | removeMark(pIndex : int) : void |
| +     | getMark(pIndex : int) : int     |
| +     | removeAllMarks() : void         |
| +     | calculateAverage() : int        |
| +     | indexOfBestMark() : int         |
| +     | indexOfMark(pMark : int) : int  |
| +     | sortAscending() : void          |

1. Créez une nouvelle classe `Marks` qui possède un seul attribut privé c.-à-d. une liste d'entiers appelée `alMarks`.
2. Ajoutez une méthode `addMark` qui permet d'ajouter une note `pMark` à la liste actuelle. On suppose sans vérification que cette note est une note valable entre 1 et 60.
3. Ajoutez une méthode `getMark` qui retourne la note à l'index indiqué en paramètre.
4. Ajoutez une méthode `removeMark` qui supprime la note à l'index spécifié.
5. Ajoutez une méthode `removeAllMarks` qui vide la liste.
6. Ajoutez une méthode `calculateAverage` qui calcule la moyenne arrondie vers le haut de toutes les notes dans la liste. Si la liste est vide la méthode retourne -1.

## Astuce

La méthode `double Math.ceil(double pN)` permet d'arrondir un nombre réel vers le haut.

7. Ajoutez la méthode `indexOfBestMark` qui retourne l'index de la meilleure note.
8. Ajoutez la méthode `indexOfMark` qui retourne l'index de la première occurrence de la note indiquée en paramètre.
9. Ajoutez la méthode `sortAscending` qui trie la liste par ordre croissant.