

---

# Compteur de mots (partie 2)

---

Dans cette partie, vous allez réaliser votre première implémentation fonctionnelle de l'outil statistique sur les textes. Cette implémentation sera réalisée à l'aide d'un tableau.

## 1 Classe de base

Dans un premier temps, créez une classe permettant de représenter un mot, ainsi que son nombre d'occurrence au sein d'un texte. Voici la structure de la classe à implémenter :

```
public class Mot {
    private String mot;
    private int cpt;
    public Mot(String m);
    public void nouvelleOccurrence();
    public int getOccurrence();
    public String getMot();
}
```

Le constructeur initialise la classe et place le compteur d'occurrence (`cpt`) à 1. La méthode `nouvelleOccurrence()` incrémente le `cpt` de 1, `getOccurrence()` retourne la valeur de `cpt`, et `getMot()` retourne le `String` « `mot` » contenu dans cet objet.

## 2 Implémentation via un tableau

Nous allons maintenant créer une classe permettant *concrètement* d'effectuer les statistiques sur le texte fourni en paramètre.

```
public class CompteurTableau extends Compteur {
    public final int TAILLE_INITIALE = 100;
    private Mot[] elements;
    public void addOccurrence(String mot) {TO DO}
    public CompteurTableau (String fichierTexte) {TO DO}
}
```

Dans cette version, `addOccurrence` doit vérifier que le tableau puisse accueillir un potentiel nouveau mot. En particulier, lors du premier appel à cette méthode, le tableau `elements` sera initialisé à une taille `TAILLE_INITIALE`. Plus tard dans l'exécution, si le tableau n'est pas assez grand pour accueillir un nouveau mot, il sera redimensionné et son

contenu copié dans le nouveau tableau créé. Par convention, quand le tableau se montre trop petit pour contenir un nouveau mot, on doublera sa taille. Il vous est conseillé :

1. de conserver le nombre courant de mots contenus dans le tableau dans une variable d'instance.
2. de vérifier dans la méthode `addOccurrence` que le tableau contient bien une place pour le mot à ajouter, et le redimensionne sinon.

### 3 Affichage des résultats

Implémentez une méthode `main()` de manière à ce que l'on puisse utiliser votre programme en ligne de commande, de la manière suivante :

```
> java CompteurTableau fichierTexte
```

Lors de cet appel, votre programme analysera le fichier donné en paramètre, et affichera un ensemble de statistiques :

- le nom du fichier concerné
- le nombre total de mots
- le nombre de mots de taille supérieure à 4
- la liste, en ordre décroissant, des 10 mots (de taille supérieure à 4) les plus fréquents, avec leur nombre d'occurrence respective. En cas d'égalité sur le(s) dernier(s) mot(s) de cette liste, les mots seront traités indifféremment (n'importe quel mot ayant ce nombre d'occurrence, à égalité, peut être utilisé).

Ces statistiques seront imprimées sur la sortie standard (*stdout*) selon un format, dont voici un exemple :

```
Fichier : exemple.txt
Nombre de mots : 6396
Nombre de mots de taille > 4 : 3721
-----
Mot les plus fréquents :
14 voirie
12 important
12 véridique
10 merveilleux
9 croyance
8 pouvoir
8 informatique
4 essentiel
3 violence
2 avenue
```

Différents fichiers permettant de tester votre programme sont disponibles sur Moodle.