

**Exercices du module**  
**Renforcement Technique JAVA**

## → Important !

Vous devrez rendre vos exercices sur un dépôt GitHub que vous aurez préalablement créé. Il devra être public et s'appeler « **JavaExercises\_[votre nom-votre prénom]** ». Par exemple : « **JavaExercises\_dupont-martin** ».

A l'intérieur de votre dépôt, vous devrez ajouter un dossier nommé « **file2** ». Chacun de vos exercices devra être dans un fichier distinct et déposé dans ce dossier. Vous nommerez les fichiers « **exercice1.java** », « **exercice2.java** », etc.

### **Rappel des commandes Git essentielles à utiliser sur votre terminal :**

- `git clone [nom de dépôt]`
- `git add [nom de fichier]`
- `git commit -m "[description de vos modifications]"`
- `git push origin main`
- `git status`
- `git log`

## ⇒ Exercice 1

Écrivez une variable appelée « **stock** » de type « **ArrayList** » de type « **Integer** » qui devra être construit avec aucun argument. À la suite de cela, vous devrez ajouter plusieurs nombres à votre liste, ces nombres sont les suivants : 14, 887, 58, 713, 179, 512, 786 et 29. Pour vous aider, vous pouvez vous rendre sur [ce lien](#).

Vous allez créer une méthode privée et statique nommé « **sum** », elle prendra en paramètre votre stock avec le type qui correspond et retournera un « **Integer** ». Dans cette méthode vous devrez additionner tous les membres de la liste et retourner le total.

Pour terminer la méthode « **main** » vous devrez faire appel à la méthode « **sum** » puis afficher le résultat dans le terminal précédé de « Contenu du stock : ».

Résultat attendu :

```
java exercisel  
Contenu du stock : 3178
```

## ⇒ Exercice 2

Vous devez, dans un premier temps, vérifier que la variable d'argument « **args** » passé en paramètre de la méthode « **main** » contient strictement une valeur à l'intérieur. Si cela n'est pas le cas, vous pouvez afficher un message d'erreur et terminer le processus en retournant une valeur vide.

A la suite, écrivez une variable appelée « **tab** » de type « **ArrayList** » de type « **String** », qui devra être construit avec aucun argument. À la suite de cela, vous devrez ajouter plusieurs fruits à votre liste, ces fruits sont les suivants : kiwi, pomme, poire, litchi et rhubarbe.

Ensuite, créez une méthode privée et statique nommé « **findFruit()** » qui aura en paramètre votre liste **tab** et votre argument « **args[0]** » de type « **String** » qui sera appelé « **str** » dans la méthode. Elle retournera un entier correspondant à l'index dans le tableau ou à « -1 » si le fruit n'est pas dans le tableau.

Il faudra utiliser une boucle **for()** pour trouver l'index du fruit. Pour chaque passage dans la boucle, effectuez une comparaison pour vérifier si les valeurs sont identiques entre l'index de la boucle et l'argument « **str** » et si c'est positif, retournez l'index. Si à la fin de la boucle **for**, vous n'avez pas trouvé de comparaison positives, retournez « -1 ».

De retour dans la méthode « **main** », vous allez devoir instancier une variable nommé « **findIndex** » qui sera le retour de votre méthode « **findFruit(...)** ». Vous ferez ensuite une comparaison afin d'afficher le bon message en fonction de l'index. Les messages d'exemple sont ci-dessous.

**Attention :** N'oubliez pas que le fruit passé en argument change et ne doit pas être le même dans vos messages !

Résultat attendu en cas de succès :

```
java exercise2 pomme
Le fruit pomme est à l'index 1
```

Résultat attendu en cas d'échec :

```
java exercise2 banane
banane n'est pas dans le tableau !
```

### ⇒ Exercice 3

Le but de cet exercice est d'afficher le nombre de voyelles contenues dans une phrase.

Écrivez une variable appelée « **quotations** » de type « **ArrayList** » de type « **String** » qui devra être construit avec aucun argument. À la suite de cela, vous devrez ajouter plusieurs nombres à votre liste, ces nombres sont les suivants : ["Quand Paris s'enrhume, l'Europe a froid", "S'il y a un diamant dans la poitrine, il brille sur le visage", " La joie de vivre n'est pas un but, mais un devoir"].

Puis, créez une méthode « **countVowels** » qui prendra en paramètres une « **String** » nommé « **str** ». Dans cette méthode, vous devrez compter puis retourner le nombre de voyelles dans la phrase « **str** ».

De retour dans la méthode « **main** » vous devrez, dans une boucle, afficher la phrase et le nombre de voyelles qu'elle contient en suivant l'exemple ci-dessous.

Résultat attendu :

```
java exercise3
Quand Paris s'enrhume, l'Europe a froid : 14
S'il y a un diamant dans la poitrine, il brille sur le visage : 21
La joie de vivre n'est pas un but, mais un devoir : 17
```