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 ».

Une fois votre dépôt cloné sur votre ordinateur, vous devrez ajouter un dossier nommé « **file3** » à l'intérieur. Chacun de vos exercices devra être dans un fichier distinct et déposé dans ce dossier. Vous nommerez les fichiers « **exercise1.java** », « **exercise2.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 boucle **for** qui va avoir en paramètre une variable « **i** » égale à **1** et dont la valeur incrémentera à chaque tour jusqu'à ce que $i soit supérieur à **100**. Dans cette boucle, vous allez devoir effectuer les conditions suivantes :

- Si le nombre est un multiple de 3, vous devrez afficher « **Fizz** »
- Si le nombre est un multiple de 5 vous devrez afficher « **Buzz** »
- Si le nombre est un multiple de 3 et 5, vous devrez afficher « **FizzBuzz** »
- Sinon, vous afficherez le nombre

Chaque fin de cycle de boucle devra être suivi d'un espace.

Résultat attendu :

```
1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz 11 Fizz 13 14 FizzBuzz 16 17 Fizz 19 Buzz Fizz 22 23
Fizz Buzz 26 Fizz 28 29 FizzBuzz 31 32 Fizz 34 Buzz Fizz 37 38 Fizz Buzz 41 Fizz 43 44 F
izzBuzz 46 47 Fizz 49 Buzz Fizz 52 53 Fizz Buzz 56 Fizz 58 59 FizzBuzz 61 62 Fizz 64 Buz
z Fizz 67 68 Fizz Buzz 71 Fizz 73 74 FizzBuzz 76 77 Fizz 79 Buzz Fizz 82 83 Fizz Buzz 86
 Fizz 88 89 FizzBuzz 91 92 Fizz 94 Buzz Fizz 97 98 Fizz Buzz
```

⇨ Exercice 2

En reprenant le code de l'exercice 1, vous allez devoir, par le biais des arguments lors de l'exécution, y définir le minimum et le maximum de votre boucle. Vous devrez y implémenter une gestion d'erreurs afin que qu'en cas de soucis un message d'erreur soit retourné.

L'exécution s'effectuera de cette manière :

java exercise2 10 120

⇨ Exercice 3

Écrivez les variables suivantes :

- « **count** » a pour valeur **0**
- « **start** » contient la valeur **1**
- « **end** » possède la valeur **50**

Dans une boucle, vous allez devoir calculer la somme des nombres pairs, entre la variable « **start** » et « **end** ». Vous vous servirez de la variable « **count** » pour stocker vos additions.

Vous afficherez ensuite cela avec un message : « La somme des nombres pairs entre 1 et 50 est : ». Vous pourrez ensuite changer la valeur de **start** et **end** pour vérifier que tout fonctionne correctement et dynamiquement.

Résultat attendu :

```
La somme des nombres pairs entre 1 et 50 est : 650
```

⇨ Exercice 4

Écrivez les variables suivantes :

- « **count** » a pour valeur **0**
- « **start** » contient la valeur **1**
- « **end** » possède la valeur **10**

Dans une boucle, vous allez devoir calculer la somme des nombres au carré, entre la variable « **start** » et « **end** ». Vous vous servirez de la variable « **count** » pour stocker vos additions.

Vous afficherez ensuite cela avec un message : « La somme des carrés des nombres entre 1 et 10 est : » ainsi que le résultat de votre calcul. Vous pourrez ensuite changer la valeur de $start et $end pour vérifier que tout fonctionne correctement et dynamiquement.

Résultat attendu :

```
La somme des des carrés des nombres entre 1 et 10 est : 385
```

⇨ Exercice 5

Dans une méthode nommée « **averageCalc** » qui prend en paramètre une variable « **numbers** », qui sera un tableau d'entier (ArrayList), et retournera un entier. Vous allez devoir effectuer le calcul de la moyenne des éléments du tableau. Pour rappel, une moyenne est calculée de cette manière :

Somme de tous les nombres du tableau / Nombre d'élément du tableau

Dans la méthode, vous aurez besoin de 2 variables : « **sum** » qui est égale à **0** et « **nbElement** » qui sera égale à la somme des éléments du tableau.

Par la suite, vous additionnerez à la variable « **sum** » tous les éléments dans une boucle **foreach**. Vous ferez ensuite le calcul de la moyenne avant de retourner le résultat avec le mot-clé « return ».

Dans la méthode « **main** » vous allez définir un tableau de nombre nommé « **nbArray** » qui contiendra les nombres suivants : **75, 80, 90, 95, 85**. Vous allez également créer une variable « **averageArray** » qui sera égale à la méthode « **averageCalc** » créée précédemment avec pour paramètre « **nbArray** ».

Résultat attendu :

```
La moyenne de la liste est : 85
```