

Labo 3 : Manipulation de Nombres avec Menu Interactif

Objectif : Créer un programme Scala qui demande à l'utilisateur d'entrer trois nombres et lui présente un menu avec plusieurs options d'opérations. Le programme doit permettre à l'utilisateur de choisir parmi plusieurs options, telles que l'addition, la multiplication, le tri, la détermination des nombres pairs et impairs, et la recherche du nombre maximum ou minimum.

Exigences

1. Entrée de Nombres :

- Demandez à l'utilisateur d'entrer trois nombres.
- Stockez ces nombres dans une liste pour pouvoir les manipuler facilement.

2. Menu d'Options :

- Présentez un menu avec les options suivantes :
 1. Additionner les trois nombres.
 2. Multiplier les trois nombres.
 3. Trier les nombres (ordre croissant).
 4. Afficher les nombres pairs et impairs.
 5. Trouver le nombre maximum et minimum.
 6. Quitter le programme.

3. Fonctionnalités :

- Chaque option doit être implémentée comme une fonction distincte.
- Utilisez une boucle pour afficher le menu jusqu'à ce que l'utilisateur choisisse de quitter.

Exemple d'Exécution :

```
Entrez le premier nombre : 5
Entrez le deuxième nombre : 2
Entrez le troisième nombre : 9

Menu :
1. Additionner les nombres
2. Multiplier les nombres
3. Trier les nombres
4. Afficher les nombres pairs et impairs
5. Trouver le nombre maximum et minimum
6. Quitter
Choisissez une option : 1
Résultat : 16

Menu :
1. Additionner les nombres
2. Multiplier les nombres
3. Trier les nombres
4. Afficher les nombres pairs et impairs
5. Trouver le nombre maximum et minimum
6. Quitter
Choisissez une option : 4
Pairs : 2
Impairs : 5, 9

Menu :
1. Additionner les nombres
2. Multiplier les nombres
3. Trier les nombres
4. Afficher les nombres pairs et impairs
5. Trouver le nombre maximum et minimum
6. Quitter
Choisissez une option : 6
Au revoir !
```

Explication des Fonctions

1. **add** : Additionne tous les nombres de la liste et retourne le résultat.
2. **multiply** : Multiplie tous les nombres de la liste et retourne le résultat.
3. **sort** : Trie les nombres dans l'ordre croissant et retourne la liste triée.
4. **separateEvenOdd** : Sépare les nombres pairs et impairs de la liste, retourne un tuple contenant deux listes (pairs et impairs).
5. **findMaxMin** : Retourne le nombre maximum et le minimum de la liste.

Critères d'Évaluation

1. **Fonctionnalité** : Chaque option du menu fonctionne correctement, et les résultats sont précis.
2. **Clarté du Code** : Utilisation correcte des fonctions, noms de variables explicites, et séparation claire des opérations.
3. **Interface Utilisateur** : Le menu est intuitif et permet une navigation facile pour l'utilisateur.
4. **Manipulation de Listes et Pattern Matching** : Bon usage de List, filter, max, min, et pattern matching.