

## TP 4 : Collections et Streams

On se propose de développer une application Java destinée à un comité d'organisation d'un tournoi sportif afin de permettre la gestion des équipes. Dans ce contexte et après une brève analyse il apparaît que :

- ✓ Un joueur est caractérisé par son nom-prénom, son âge, sa taille, son poids, le nombre de matchs joués, et son rôle dans l'équipe affecté à partir de cet ensemble statique {Gardien, Défense, Centre, Attaque}
- ✓ Une équipe est composée par une liste de joueurs, et possède un ensemble de titres
- ✓ Un tournoi est caractérisé par un lieu et un ensemble des équipes. A chaque équipe on attribue un identifiant numérique unique lors du tournoi.
  - 1) Implémenter les classes Joueur et Titre ainsi que les méthodes que vous jugez nécessaires pour chaque classe.
  - 2) Implémenter la classe Equipe. Il est à noter qu'on doit avoir ces méthodes :
    - ✓ ajouterJoueur(Joueur)
    - ✓ ajouterTitre(Titre)
    - ✓ getJoueur(int)
    - ✓ getJoueurs()
    - ✓ afficher() : affichage de toutes les données avec des itérateurs
    - ✓ affichJoueurs() avec méthode forEach()
    - ✓ afficherTitres() avec méthode forEach()
    - ✓ redéfinition de toString() : utiliser StringBuffer et une boucle forEach
  - 3) Implémenter une classe Test possédant une méthode main() afin de créer une équipe, d'ajouter des joueurs et des titres et de tester les affichages.
  - 4) Implémenter l'interface fonctionnelle Comparable afin de comparer les âges des joueurs.
  - 5) Implémenter des méthodes statiques générant des comparateurs des tailles, et aussi des poids des joueurs. Pour la comparaison des matchs joués, créer une classe indépendante implémentant l'interface Comparator.
  - 6) Effectuer dans la classe principale:
    - ✓ Le tri par ordre croissant des matchs joués avec affichage.
    - ✓ Le tri par taille croissante des joueurs. Utiliser un stream pour l'affichage.
    - ✓ Le tri par poids croissant. Utiliser la méthode sort() pour les listes et une expression Lambda

7) Filtrer les joueurs de taille plus de 180 cm, puis les trier par ordre de taille croissante, et finalement afficher une seule ligne du type  
180 : Ali, 183 : Fred, 195 : Bob

8) Ajouter dans la classe Joueur les méthodes suivantes :

```
public Object valeur(int i) {  
    switch (i) {  
        case 0: return this.getNomPrenom();  
        case 1: return this.getTaille();  
        case 2: return this.getPoids();  
        case 3: return this.getNbrMatch();  
        case 4: return this.getRole();  
        default: throw new RuntimeException("pas de champ pour cet indice");  
    }  
}
```

```
public Stream<Object> valeurs(){  
    return Stream.iterate(0, n->n+1).limit(5).map(n -> this.valeur(n));  
}
```

Expliquer-les et tester dans la classe principale.

9) Implémenter la classe Tournoi ainsi que les méthodes que vous jugez nécessaires pour cette classe.