

Les Masters

On ajoute un nouveau type de compétition, les **Master**, qui font s'affronter les compétiteurs en deux phases. La compétition commence par une phase de poules à l'issue de laquelle certains compétiteurs sont sélectionnés pour disputer la phase finale du tournoi. Les poules sont organisées sous forme de championnat tandis que la phase finale se déroule sous la forme d'un tournoi à élimination directe.

La méthode de sélection des compétiteurs retenus pour disputer la phase finale du Master est précisée à la création du Master, et peut varier d'un Master à un autre. Par exemple, une première compétition de type Master peut répartir 16 joueurs en 4 poules de 4 joueurs et ne conserver que les meilleurs de chaque poule pour la seconde phase. Un second Master peut lui considérer 24 compétiteurs répartis en 3 groupes de 8, pour ne sélectionner pour la phase finale que les 2 premiers de chaque poule auxquels s'ajoutent les 2 meilleurs troisièmes, toutes poules confondues. Bien entendu, d'autres méthodes pour sélectionner les compétiteurs entre la première et la seconde phase peuvent être envisagées : deux premiers de chaque groupe, meilleurs seconds, repêchage du dernier de chaque poule, etc. Quelle que soit la méthode de sélection utilisée, le nombre de compétiteurs accédant en phase finale doit être une puissance de 2.

Le vainqueur du Master est le compétiteur finissant vainqueur de la phase finale. Ce n'est donc pas nécessairement le compétiteur qui a le plus de victoires qui remporte le tournoi, puisque un compétiteur peut terminer premier de sa poule en ayant moins de points que le premier d'une autre poule. L'affichage du classement d'un Master fait donc apparaître le classement de chacune des poules, ainsi que le classement de la phase finale.

Journalistes et Bookmakers

Les compétitions sont désormais si populaires que les média et les parieurs s'y intéressent. Des journalistes peuvent donc assister aux compétitions et diffuser les résultats des matchs. On considère que la diffusion d'un résultat se fait par l'affichage dudit résultat dans la sortie standard. Les bookmakers, quant à eux, font évoluer les cotes des compétiteurs en fonction des résultats des matchs de la compétition. Ils maintiennent une liste des cotes des compétiteurs qu'ils affichent et font évoluer en fonction des victoires (diminution de la cote) et des défaites (augmentation de la cote). Par exemple, "Victoire de Léa (cote 4) face à Audrey (cote 1). La cote de Léa passe à 3, celle d'Audrey à 2". Lorsqu'un journaliste ou un bookmaker vient assister à une compétition, il assiste à tous les matchs de cette dernière.

Q 1 . En vous inspirant des fichiers fournis en annexe, proposez une conception, un programme et la suite de tests associée qui reprennent l'ensemble des fonctionnalités présentées dans ce sujet.

Annexes - La classe `util.MapUtil`

Cette classe utilitaire propose une méthode permettant de classer une `Map<Clé, Valeur>` dans l'ordre décroissant des valeurs, à condition que ces valeurs soient des entiers. Le fichier est disponible sur le portail.

```
package util;
```

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
```

```
public class MapUtil {
```

```
    public static <K, V extends Comparable<? super V>> Map<K, V> sortByDescendingValue(Map<K, V> map) {
```

```
        List<Entry<K, V>> sortedEntries = new ArrayList<Entry<K, V>>(map.entrySet());
```

```
        Collections.sort(sortedEntries, new Comparator<Entry<K, V>>() {
```

```
            @Override
```

```
                public int compare(Entry<K, V> e1, Entry<K, V> e2) {
```

```
                    return e2.getValue().compareTo(e1.getValue());
```

```
                }
```

```
            });
```

```
        Map<K, V> result = new LinkedHashMap<>();
```

```
        for (Entry<K, V> entry : sortedEntries) {
```

```
            result.put(entry.getKey(), entry.getValue());
```

```
        }
```

```
        return result;
```

```
    }
```

```
}
```

Un exemple de trace d'exécution

Blastoff vs Drift --> Blastoff wins!
Blastoff vs Lynx --> Lynx wins!
Blastoff vs Catalyst --> Catalyst wins!
Blastoff vs Raven --> Raven wins!
Blastoff vs Midas --> Midas wins!
Drift vs Blastoff --> Blastoff wins!
Drift vs Lynx --> Lynx wins!
Drift vs Catalyst --> Drift wins!
Drift vs Raven --> Drift wins!
Drift vs Midas --> Midas wins!
Lynx vs Blastoff --> Lynx wins!
Lynx vs Drift --> Drift wins!
Lynx vs Catalyst --> Lynx wins!
Lynx vs Raven --> Lynx wins!
Lynx vs Midas --> Lynx wins!
Catalyst vs Blastoff --> Catalyst wins!
Catalyst vs Drift --> Drift wins!
Catalyst vs Lynx --> Lynx wins!
Catalyst vs Raven --> Catalyst wins!
Catalyst vs Midas --> Midas wins!
Raven vs Blastoff --> Raven wins!
Raven vs Drift --> Drift wins!
Raven vs Lynx --> Raven wins!
Raven vs Catalyst --> Raven wins!
Raven vs Midas --> Raven wins!
Midas vs Blastoff --> Blastoff wins!
Midas vs Drift --> Midas wins!
Midas vs Lynx --> Lynx wins!
Midas vs Catalyst --> Catalyst wins!
Midas vs Raven --> Raven wins!

*** Ranking ***

Lynx - 8
Raven - 6
Drift - 5
Catalyst - 4
Midas - 4
Blastoff - 3