TP 2 - Multithread

Le but de ce TP est de manipuler les processus légers (threads) en Java. Pour cela, nous allons simuler une application qui agrège les commentaires de plusieurs matchs de foot se déroulant en même temps, afin d'obtenir quelque chose qui ressemble à ceci :

```
Paris-Marseille : Faute des locaux
Nîmes-Lyon : Pénalty pour les locaux
Paris-Marseille : Coup-franc des visiteurs
Montpellier-Dijon : Touche en faveur des locaux
Paris-Marseille : Belle occasion pour les visiteurs !
Nîmes-Lyon : But des locaux !!!
```

1/ Récupérez la classe **`CommentGenerator`** qui contient une liste de commentaires. Écrivez le corps de la méthode **`getRandomComment()`** qui permet d'en générer un au hasard en utilisant la méthode **`Math.random()`**

2/ Créez une classe **`ThreadMatch`** qui a comme attribut le nom des deux équipes, avec un constructeur par paramètres.
Cette classe implémente l'interface **`Runnable`**, et doit donc implémenter la méthode **`run()`**. Dans cette dernière, écrivez une boucle infinie dans laquelle vous générez un commentaire, puis mettez en attente le thread pendant un temps aléatoire entre 0 et 10 secondes.

3/ Dans le main, créez trois threads **`ThreadMatch`** qui s'exécutent simultanément.

Questions supplémentaires :

4/ Utilisez la méthode **`join()`** afin de faire débuter un nouveau match à la fin d'un autre (vous aurez avant cela changé la boucle infinie de chaque thread en une boucle qui affiche par exemple 3 commentaires et s'arrête). On voudra par exemple obtenir quelque chose comme cela :

```
Début du match Paris - Marseille
Paris-Marseille : Touche en faveur des locaux
Paris-Marseille : Hors-jeu des visiteurs
Paris-Marseille : Faute des visiteurs
fin du match entre Paris et Marseille
Début du match Nîmes - Lyon
Nîmes-Lyon : Pénalty pour les locaux
Nîmes-Lyon : Envahissement du terrain par des supporters !
```

5/ Modifiez les classes pour afficher et mémoriser le score d'un match, dès qu'un but a été marqué.

6/ Modifiez la classe **`CommentGenerator`** afin qu'il implémente le patron de conception **Singleton**