

# 1.2. Les collections : TP Cyclistes, les Map

JAVA AVANCE

Etape 2

(IEF Christophe Louër)

<<Java Class>>	
Epreuve	
model entities	
id: UUID	
nom: String	
dateDebut: LocalDate	
dateFin: LocalDate	
Epreuve(String,LocalDate,LocalDate)	
getId():UUID	
getNom():String	
getDateDebut():LocalDate	
setDateDebut(LocalDate):void	
getDateFin():LocalDate	
setDateFin(LocalDate):void	
Epreuve()	

## 1. Classe Epreuve

Ecrire la classe *Epreuve* listant les épreuves du calendrier cycliste.  
L'identité est définie sur l'attribut *id*.

Pas de contraintes particulières.

Ecrire 2 fabriques :

```
public static Epreuve fabriquerEpreuve(String nom,LocalDate dateDebut,LocalDate dateFin)
```

```
public static Epreuve fabriquerEpreuve(String nom,LocalDate dateDebut)
```

Pour la dernière, la date de fin est la même que la date de début : c'est une course classique.

## 2. La classe Cycliste :

Ajouter l'attribut *palmares* de type *Map* dont la clé est une épreuve et la valeur le classement du cycliste.

Ecrire la méthode `public void modifierPalmares(Epreuve e,int classement)` qui permet d'ajouter ou modifier un palmarès pour une épreuve.

Vous testez que l'épreuve en paramètre est attachée et que le classement est supérieur ou égal à zéro. Dans ces deux cas vous lèverez la runtime exception *CyclisteRuntimeException*.

### 3. La classe DemoCollection :

Ajouter les méthodes suivantes préparant des epreuves :

```
public static Stream<Epreuve> creerParisNice() {
    return Stream.of(EntitiesFactory.fabriquerEpreuve("PARIS-NICE", LocalDate.of(2019, Month.MARCH, 10),
LocalDate.of(2019, Month.MARCH, 13)),
                    EntitiesFactory.fabriquerEpreuve("PARIS-NICE", LocalDate.of(2018,
Month.MARCH, 4), LocalDate.of(2018, Month.MARCH, 11)),
                    EntitiesFactory.fabriquerEpreuve("PARIS-NICE", LocalDate.of(2017,
Month.MARCH, 5), LocalDate.of(2017, Month.MARCH, 12)),
                    EntitiesFactory.fabriquerEpreuve("PARIS-NICE", LocalDate.of(2016,
Month.MARCH, 6), LocalDate.of(2016, Month.MARCH, 13)));
}
public static Stream<Epreuve> creerParisRoubaix() {
    return Stream.of(EntitiesFactory.fabriquerEpreuve("PARIS-ROUBAIX", LocalDate.of(2019, Month.APRIL, 14)),
                    EntitiesFactory.fabriquerEpreuve("PARIS-ROUBAIX", LocalDate.of(2018,
Month.APRIL, 8)),
                    EntitiesFactory.fabriquerEpreuve("PARIS-ROUBAIX", LocalDate.of(2017,
Month.APRIL, 9)),
                    EntitiesFactory.fabriquerEpreuve("PARIS-ROUBAIX", LocalDate.of(2016,
Month.APRIL, 10)));
}
public static Stream<Epreuve> creerCriteriumDauphine() {
    return Stream.of(EntitiesFactory.fabriquerEpreuve("CRITERIUM DU DAUPHINE", LocalDate.of(2019, Month.JUNE,
9), LocalDate.of(2019, Month.JUNE, 16)),
                    EntitiesFactory.fabriquerEpreuve("CRITERIUM DU DAUPHINE",
LocalDate.of(2018, Month.JUNE, 3), LocalDate.of(2018, Month.JUNE, 10)),
                    EntitiesFactory.fabriquerEpreuve("CRITERIUM DU DAUPHINE",
LocalDate.of(2017, Month.JUNE, 4), LocalDate.of(2017, Month.JUNE, 11)),
                    EntitiesFactory.fabriquerEpreuve("CRITERIUM DU DAUPHINE",
LocalDate.of(2016, Month.JUNE, 5), LocalDate.of(2016, Month.JUNE, 12)));
}
public static Stream<Epreuve> creerTourFrance() {
    return Stream.of(EntitiesFactory.fabriquerEpreuve("TOUR DE FRANCE", LocalDate.of(2019, Month.JULY, 6),
LocalDate.of(2019, Month.JULY, 28)),
                    EntitiesFactory.fabriquerEpreuve("TOUR DE FRANCE", LocalDate.of(2018,
Month.JULY, 7), LocalDate.of(2018, Month.JULY, 29)),
                    EntitiesFactory.fabriquerEpreuve("TOUR DE FRANCE", LocalDate.of(2017,
Month.JULY, 1), LocalDate.of(2017, Month.JULY, 23)),
                    EntitiesFactory.fabriquerEpreuve("TOUR DE FRANCE", LocalDate.of(2016,
Month.JULY, 2), LocalDate.of(2016, Month.JULY, 24)));
}
```

Enfin écrire une méthode ajoutant à chaque cycliste d'une équipe un palmarès pour chaque épreuve en calculant aléatoirement son classement.

Vous afficherez les cyclistes de l'équipe avec le palmarès.