

Atelier 7 : location de vélo

Estimé : long

Objectif :

1. La location

Créez les classes Location et Client.

Un client est représenté par :

- Son nom
- Son prénom
- Son Adresse
- La liste de ses locations en cours

Une location est représentée par :

- Une date de début (LocalDateTime)
- Une date de fin (LocalDateTime)
- Le client à l'origine de la location
- Un article (cycle)
- Un booléen indiquant si la facture a été réglée ou non

La classe Location propose les méthodes suivantes :

Manque des méthodes volontairement

- terminerLocation() : affecte la date du jour à l'attribut date de fin
- calculerMontant() : calcule le nombre d'heures écoulées entre le début de la location et la fin, et retourne le prix total. Si la date de fin n'est pas renseignée, on utilise la date du jour. Vous pouvez utiliser l'exemple suivant pour déterminer le nombre de jours entre deux dates :

```
LocalDateTime dateTime1 = LocalDateTime.now();
LocalDateTime dateTime2 = LocalDateTime.of(2022, 02, 10, 12, 30);
System.out.println(ChronoUnit.HOURS.between(dateTime2, dateTime1));
```

La classe Location dispose de deux constructeurs :

- Le constructeur de base prendra le client et l'article en paramètre. La date de départ sera initialisée à la date du jour, tandis que la date de fin ne sera pas renseignée.
- Un constructeur de copie, qui prendra une autre Location en paramètre. Elle copie l'article et le client utilisé, mais suit les mêmes règles que le constructeur de base pour les dates.

2. Améliorer le magasin

A la fin de l'atelier 6, nous disposons d'un Magasin manipulant un tableau de cycles. Améliorez cette structure en utilisant une des Collections vues en cours.

Ajoutez les méthodes suivantes à vos cycles :

- enregistrerArticle() : prend un cycle en paramètre et l'ajoute à la liste des cycles disponibles à la location

- `chercherArticle()` : prend un nom de modèle en paramètre, et retourne le cycle dont le modèle correspond
- `supprimerArticle()` : prend un cycle en paramètre, et le supprime de la liste des articles disponibles à la location
- `consulterLocationsEnCours()` : retourne la liste des locations ne possédant pas de date de fin de location

De plus, le gérant du magasin souhaite pouvoir consulter facilement ses clients en tapant leur nom de famille.

Ajoutez un attribut au magasin, qui permette de retrouver rapidement une instance de client grâce à son nom. (→ pour quelle Collection optez-vous dans ce cas de figure ?)

Ajoutez les méthodes suivantes au Magasin :

- `rechercherClient()` : prend un nom en paramètre, et renvoie le client correspondant. On considère qu'il n'y a pas d'homonymes. (*Pour aller plus loin : la méthode `rechercherClient` peut retourner une liste de clients pour gérer les homonymes*)
- `calculerMontant()` : prend un client en paramètre, et détermine le montant total de toutes ses location non-réglées.

Rédigez la classe de test vous permettant de manipuler des cycles, des clients, des factures et des magasins, afin de vérifier le bon fonctionnement de ce que vous avez développé.