



## Premiers pas avec l'héritage en Java

### La poste en Laponie - 2e partie

Dans ce sujet, nous retravaillons certains des points du précédent TP sur les objets postaux.

## 1 Tarif de remboursement

Nous allons mettre en place des méthodes `double tarifRemboursement()` pour calculer le tarif de remboursement d'un objet postal, d'une lettre ou d'un colis. Ce tarif est remboursé au client en cas de perte de l'objet postal.

**Question 1** Etudiez le calcul du tarif de remboursement des lettres et des colis pour repérer s'il y a quelque chose de commun à tous les objets postaux et le placer dans la méthode `double tarifRemboursement()` de la classe `ObjetPostal`. Puis définissez cette méthode dans toutes les classes.

Le tarif de remboursement d'une *lettre* est de :

- 0 euro si le taux de recommandation est égal à 0,
- 1.5 euros si le taux est égal à 1,
- 15 euros si le taux est égal à 2.

Le tarif de remboursement d'un *colis* est de :

- 10% de la valeur déclarée si le taux de recommandation est 1,
- 50% de la valeur déclarée si le taux de recommandation est 2,
- 0 euros si le taux de recommandation est 0.

**Question 2** Testez la méthode sur tous les objets créés dans le `main`.

## 2 Retour sur le calcul du tarif d'affranchissement

Nous retravaillons les méthodes `double tarifAffranchissement()` qui calculent le tarif d'affranchissement d'un objet postal, d'une lettre ou d'un colis.

On rappelle que le tarif d'affranchissement d'une *lettre* se calcule de la manière suivante. Le tarif de base est de 0.5 euro auquel s'ajoutent cumulativement :

- 0.5 euro lorsque le taux de recommandation est 1 ; 1.5 euros lorsque le taux de recommandation est 2,
- 0.30 euro si c'est une lettre urgente.

Le tarif d'affranchissement d'un *colis*, de son côté, s'obtient par cumul des sommes suivantes :

- 2 euros dans tous les cas (tarif de base),
- 0.5 euro si le taux de recommandation est 1 ; 1.5 euros si le taux de recommandation est 2 (comme pour les lettres),
- 3 euros de surtaxe si le colis dépasse  $1/8$  de  $m^3$ .

### Question 3

- Définissez une méthode `tarifBase` pour les objets postaux, les lettres et les colis. Comme le tarif de base n'est pas le même suivant les objets postaux (c'est 0,5 pour les lettres et 2 pour les colis), ce doit être une méthode abstraite dans `ObjetPostal`. Quelles en sont les conséquences ? Mettez à jour ce qui doit l'être dans vos précédents programmes.

- Réétudiez le calcul du tarif d'affranchissement des lettres et des colis et réécrivez la méthode `tarifAffranchissement` pour la classe `ObjetPostal`, en utilisant `tarifBase`.
- Puis redéfinissez les méthodes nécessaires dans les deux sous-classes.

**Question 4** Testez la méthode sur tous les objets créés dans le `main`.

### 3 Sac Postal

Nous définissons à présent des sacs contenant des objets postaux. Ecrivez une classe `SacPostal` et complétez votre `main` de manière à tester vos méthodes au fur et à mesure.

#### 3.1 Structure (attributs), constructeurs, accesseurs

Un *sac postal* peut contenir un certain nombre d'objets postaux, et dispose d'une capacité maximale. Cette capacité est de  $0,5m^3$  pour un sac ordinaire. On peut fabriquer des sacs d'une autre capacité, sur demande précisant la capacité voulue.

**Question 5** Ecrivez la classe, les attributs, prévoyez les constructeurs et les accesseurs.

#### 3.2 Comportement (méthodes)

**Question 6** Ecrivez les méthodes nécessaires à la mise en place des opérations suivantes :

- On doit pouvoir ajouter un objet dans un sac, s'il y rentre.
- On veut connaître le volume occupé par un sac (compter forfaitairement  $5dm^3$  pour la toile du sac),
- On veut connaître la valeur de remboursement en cas de perte.
- Enfin, on désire aussi une méthode pour créer, remplir et retourner un nouveau sac avec tous les objets de même code postal extraits d'un autre.

**Question 7** Testez vos méthodes en créant et en manipulant un sac postal.