

TD JAVA n°1

UNE PERSONNE

Créez une classe Personne.

- Une personne est caractérisée par son nom, son prénom, son adresse, son numéro de téléphone, représentés par des chaînes de caractères.

- Donner Le code de la classe Personne en Java

- Créez un constructeur qui affecte les valeurs qui lui sont transmises (arguments du constructeur) aux attributs de l'objet en cours de création.

- Compléter le code de la classe avec le constructeur

- Donner le code pour créer une instance de la classe Personne

- Créez un constructeur de recopie.

- on peut utiliser un constructeur de recopie si on veut par exemple créer le frère d'une Personne
- 1) En utilisant le constructeur de recopie précédent et en changeant ensuite le prénom. A faire
- 2) En faisant un constructeur proche du constructeur de recopie mais dans lequel on prend en compte le nouveau prénom. A faire

UN WAGON

Un wagon est caractérisé par un tableau de personnes et par le nombre de places occupées (initialement 0). Une place est réservée si on lui a associé une personne.

- on spécifiera le nombre de places disponibles dans le wagon lors de la création de l'objet ;
- on prévoira une méthode *placeLibre()* qui renverra *false* si aucune place n'est disponible (c'est-à-dire lorsque toutes les places sont déjà occupées).
- on prévoira une méthode *affectePlace(Personne p)* qui affecte à une personne la première place libre dans le wagon.

- Proposer un diagramme UML pour la classe Wagon en spécifiant la liaison avec la classe Personne.

- Ecrire le constructeur de la classe Wagon :

- Ecrire le code de la méthode `placeLibre()`, puis celui de la méthode `affectePlace(Personne p)`.
 ➤ Poursuivre le programme principal de la question précédente en créant un wagon et en réservant une place de ce wagon pour une des deux personnes préalablement créées.

UN TRAIN

Dans un premier temps, on considère un train comme un ensemble de 6 wagons de 60 places chacun (ces deux valeurs sont fixes). Lorsqu'une personne souhaite réserver une place dans un train, on cherche le premier wagon disposant d'une place libre.

- Ecrire le constructeur de la classe Train (il contiendra un tableau de wagons nommé train).

Dans la classe Train la méthode `affectePlace(Personne)` ci-après utilise une méthode `placeLibre()` de la classe Train.

```
public void affectePlace (Personne p) {
    int wagon_disponible = placeLibre();
    if (wagon_disponible < train.length)
        train[wagon_disponible].affectePlace(p);
}
```

- Ecrire la méthode `placeLibre()` de la classe Train.
 ➤ Donner le diagramme UML de la classe Train, en n'oubliant pas de préciser le type de liaison avec la classe Wagon.

Le modèle de train prévu dans la question 3 est un peu simpliste. En fait un train est constitué des éléments suivants :

- une locomotive (0)
 ➤ un wagon de 1^{ière} classe (40)
 ➤ un wagon de 1^{ière} classe (40)
 ➤ un wagon de 2^{ième} classe (60)
 ➤ un wagon de 2^{ième} classe (60)
 ➤ un wagon de 2^{ième} classe (60)

Le nombre entre parenthèses stipule le nombre de places disponibles. On considérera dans cet exemple simplifié qu'une locomotive est un wagon dans lequel il n'y a pas de place passager.

Pour mettre en œuvre cette solution on va utiliser des méthodes de classe pour créer des locomotives, wagon de 1^{ère} classe et wagons de 2^{ème} classe.

➤ Donner :

- le nouveau diagramme UML de la classe Wagon.
- le code Java des méthodes de classe.
- le constructeur de Train.