



Modalités

- ✓ Durée : 2h.
- ✓ Vous pouvez utiliser une feuille de notes manuscrites (écrites à la main) non photocopée recto-verso de taille quelconque.
- ✓ Rendez l'annexe jointe si vous l'utilisez. Elle compte pour un intercalaire.
- ✓ Dans l'ensemble du devoir, les collections (listes, ensembles, suites...) peuvent être codées au choix par des tableaux ou des ArrayList.

I. Situation préliminaire

Il s'agit de gérer les repas des clients pour des restaurants.

La **carte** du restaurant répertorie la **liste de tous les plats** qu'un client peut commander (Le client ne sera pas représenté). Elle a aussi un nom (Par exemple, **Carte de saison**).

Plus précisément, un plat comprend un **nom** (exemple : **Jambon à la broche**, **Homard thermidor**, **Friture de joels**, **Crème brûlée à la myrte...**), sa **catégorie**, la **carte** à laquelle il appartient, et une **disponibilité** — qui exprime qu'il n'y a pas de rupture d'approvisionnement pour le plat ou l'un de ses constituants. Chaque plat a aussi un **prix** de vente.

Un **table** réunit un ou plusieurs clients ; elle a un **numéro** (ex. : **table 3**). À chaque table, est associée la **liste des commandes** des clients de la table. Une **commande** est un **plat** dans une **quantité** donnée (ex. : 2 jambons à la broche ou une friture de joels...). Le **prix** associé à une commande est le **prix** du plat fois la **quantité** commandée, le **prix** associé à une table est la somme des **prix** des commandes des clients de la table ; il constitue la **note** qui devra être payée par cette table.

Questions

1. Faites figurer dans le premier diagramme de l'annexe les relations entre classes qui rendent compte de l'énoncé.
2. Ajoutez à ce diagramme les déclarations des attributs nécessaires (Les collections pourront être représentées par des tableaux).
3. En supposant que les accesseurs utiles soient déjà écrits, écrivez la méthode `getPrix()` des classes `Plat`, `Commande` et `Table`.
4. Écrivez le constructeur standard de la classe `Plat` qui est destiné à être appelé par une instance de la classe `Carte` (Un nouveau plat sera ajouté à une carte donnée). Un nouveau plat est considéré au départ comme disponible.
5. On suppose que la classe `Carte` contient une méthode d'instance : `ajoute(Plat p)` qui ajoute un plat existant à la carte. Par exemple, l'appel à cette méthode peut être effectuée comme suit :

```
Carte c = new Carte("Carte de saison");  
c.ajoute(new Plat("Homard Thermidor", "Plat principal", c, 32));
```

Écrivez dans la classe `Carte` la méthode d'instance de signature

`ajoute(String nom, String categorie, double prix)`

qui permet d'ajouter un nouveau plat à la carte à partir de ses caractéristiques. Avec la même instance de `Carte` référencée par la variable `c`, cette méthode est utilisée comme suit :

```
c.ajoute("Homard Thermidor", "Plat principal", 32);
```

6. Écrivez les méthodes `toString` de :

- ✓ `Plat`. Renvoie la catégorie et l'intitulé du plat suivi de son prix ou de la mention **indisponible** si le plat est en rupture. Par exemple, **Jambon à la broche : 18.0€, ou Homard Thermidor : indisponible**.

- ✓ `Commande`. Restitue une chaîne de caractères qui indique les caractéristiques de la commande. Par exemple, **Jambon à la broche : 18.0€ x 2 >> 36.0€** ou **Salade de poivrons façon méchouia : 8.0€ x 1 >> 8.0€**.

- ✓ `Table`. Restitue une chaîne de caractères qui décrit la note d'une table. Par exemple :

```
Salade de poivrons façon méchouia : 8.0 x 1 >> 8.0 €  
Briç à l'oeuf sauce meurette : 9.0 x 3 >> 27.0 €  
Fleur de sureau à la croque au sel et pain d'épice : 10.0 x 2 >> 20.0 €  
Jambon à la broche : 18.0 x 2 >> 36.0 €  
Crème brûlée à la myrte : 7.0 x 3 >> 21.0 €  
-----  
Note table 3 : 112.0€
```

II. Situation élaborée

Dans la situation précédente, les clients ne pouvaient commander qu'à la carte. On souhaite désormais qu'ils puissent aussi choisir dans un menu pour un prix forfaitaire.

Dans ce cadre, un **menu** est une **carte particulière**, qui contient elle aussi un nom et une **liste de plats**. Mais on lui associe en plus un **prix**.

La **commande au menu** d'un client d'une table consiste à choisir un plat parmi ceux du menu. Elle est définie par le **menu** auquel elle réfère, par la **liste de plats** choisis, et par son **prix**, qui est celui du menu. Il y a alors deux types de **commandes** :

- ✓ les **commandes simples** : les commandes de la situation précédente,
- ✓ les **commandes au menu**, décrites ci-dessus.

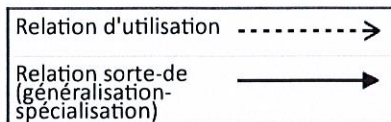
La liste des commandes d'une table peut contenir à la fois des commandes simples et des commandes au menu.

Questions

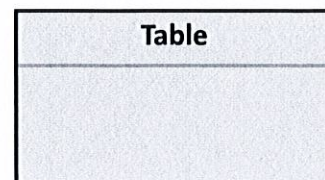
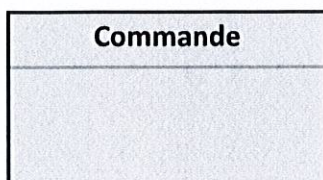
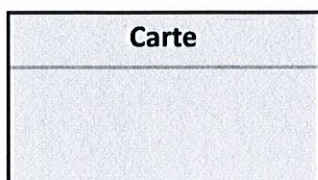
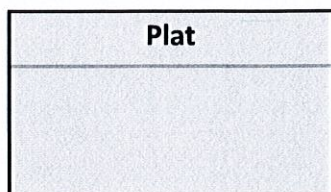
1. Faites figurer dans le deuxième diagramme de l'annexe les relations entre classes qui rendent compte de ces nouvelles modalités.
2. Ajoutez à ce diagramme les déclarations des attributs nécessaires pour rendre compte de toutes les informations de l'énoncé dans cette nouvelle forme.
3. Écrivez à nouveau la méthode `getPrix()` pour les classes `CommandeSimple`, `CommandeAuMenu`, `Commande` et `Table`. Pour ce faire, utilisez le polymorphisme hiérarchique en expliquant comment vous employez cette notion.

Annexe à rendre

Légende



I. Situation préliminaire



II. Situation élaborée

