

Projet gestion de restaurant

On vous demande de prendre en charge la réalisation d'une application en langage Java destinée à un restaurant. L'idée générale est de permettre aisément aux managers et aux serveurs d'enregistrer les menus et les plats proposés, de prendre les commandes des clients et de calculer le prix en fonction de certaines règles. Pour cela, vous répondrez aux questions suivantes :

1. Pour commencer, dessinez lisiblement sur une feuille de papier, sur laquelle figurent votre nom et prénom, le diagramme des classes Java qui correspondent aux spécifications suivantes :
 - La première classe importante est celle de Client. Un client comporte des champs spécifiques (nom, nombre de personnes, date de réservation) et des méthodes. En particulier, un client peut commander des plats et/ou des menus (éventuellement plusieurs fois le même). Un client peut émettre une réclamation. A la fin, il peut demander l'addition qui calcule la somme qu'il doit payer à l'établissement. Attention : considérez qu' "un" client correspond en réalité à plusieurs personnes installées à une même table (par exemple M. Kashluthin et son épouse).
 - Une autre classe est celle de Plat. Un plat est d'un certain type (entrée, plat de résistance, dessert, boisson) et possède un nom (comme "salade composée" ou "andouillette à la moutarde"). Il possède un prix fixe de base (par exemple 5,5 euros). Un plat est associé à des méthodes, comme promotion qui retourne la valeur vraie lorsqu'une promotion est en cours (la promotion par défaut pour tous les plats est de 20%).
 - Une autre classe correspond au Menu. Un menu contient plusieurs plats et coûte un certain prix. Le prix dépend des plats qui le composent, prix auquel on applique une réduction spécifique à chaque menu (par exemple : 10% ou 15%). On peut ajouter n'importe quel plat au menu avec la méthode ajouter. — Une classe ClientCE permet d'identifier un certain type de client. Cette classe hérite de Client et ajoute des caractéristiques spécifiques. Ainsi, le client qui vient grâce à son Comité d'Entreprise (CE) peut avoir des réductions sur l'addition finale. La réduction est un montant fixe en euros qui dépend du nombre de personnes qui viennent manger (par ex. 2,5 euros si la personne est seule, 8 euros s'ils sont à deux, etc.).
 - Enfin, la classe principale (celle qui contient le main) est la classe Restaurant. Prévoyez des champs comme un nom, une adresse, un numéro de téléphone, un numéro SIRET.
2. Une fois le schéma réalisé, mettez la feuille de côté et exécutez l'environnement de développement Java (Eclipse ou NetBeans). Dans ce qui suit, il vous est demandé de créer les classes que nous venons de décrire afin d'aboutir à une petite application fonctionnelle. N'hésitez pas à ajouter les champs et les méthodes dont vous aurez besoin pour que cela fonctionne, même si ce n'est pas explicitement demandé dans l'énoncé (par exemple les constructeurs, les accesseurs et mutateurs nécessaires, etc.).
3. N'oubliez surtout pas de commenter votre code.
4. Créez un nouveau projet intitulé "Votre nom " et commencez par ajouter la classe principale Restaurant. Prévoyez plusieurs collections pour gérer la liste des clients, des plats et des menus. Pour gérer les plats et les menus, vous pouvez vous contenter d'utiliser un Vector ou un ArrayList. Pour les clients en revanche, il est obligatoire d'avoir recours à un TreeSet. Vous terminerez l'implémentation de cette classe ultérieurement.
- 5.

Créez la classe Plat et commencez son implémentation. En particulier, remplissez les champs qui ont été décrits au début du sujet (par défaut, pas de promotion). N'oubliez pas que le prix retourné doit prendre en compte la promotion si cette option est activée, avec une valeur identique pour tous les plats concernés. Surchargez la méthode toString afin de pouvoir afficher le plat. Master 1ère année : 18 Avril 2016 – ITIS 2011 Responsable du cours : GUERFI SOUHILA

5. Créez à présent la classe Menu et commencez son implémentation. En particulier, remplissez les champs qui ont été décrits au début du sujet. N'oubliez pas de préciser un constructeur et le corps de la méthode ajouter(Plat).

6. Ajoutez une méthode privée calcule prix qui se base sur le prix des plats qui le composent. Le prix doit se mettre à jour automatiquement à chaque fois qu'on ajoute un plat au menu. N'oubliez pas de prendre en compte la réduction spécifique au menu (c'est un pourcentage à déduire).

7. Surchargez la méthode toString afin de pouvoir afficher par la suite la composition d'un menu lorsque cela sera nécessaire.

8. Créez la classe Client avec les champs décrits plus haut. N'oubliez pas d'y ajouter la liste des menus et des plats commandés, avec le prix final de l'addition. Pour la date de réservation, vous pouvez vous contenter du jour et utiliser des valeurs par défaut pour le mois et l'année.

9. Ajoutez à Client deux méthodes commander : une pour ajouter un menu et une autre pour ajouter un plat simple.

10. Ecrivez une méthode addition qui calcule le prix total que doit payer le client à la fin de son repas.

11. Surchargez la méthode toString de Client afin d'afficher un résumé des plats commandés, avec l'addition finale.

12. Ecrivez la méthode main dans Restaurant afin de tester votre programme. Vous créerez plusieurs plats, dont un au moins avec une promotion, et 2 menus de votre choix. Créez enfin 2 objets clients et faites-les commander un menu de votre choix.

13. Créez une nouvelle classe ClientCE qui hérite de la classe Client. Cette classe possède un nouveau champ qui précise le montant de la réduction (en euros) qui s'applique à l'addition finale. N'oubliez pas que ce montant dépend du nombre de personnes à la table (à vous de décider des montants). Utilisez de préférence un switch pour instancier cette valeur. Il est également nécessaire de surcharger / outrepasser la méthode qui calcule l'addition que vous avez implémentée précédemment.

14. Dans certains cas (rares), la réduction pourrait amener à une addition négative. Cela n'est pas possible : faites en sorte de déclencher une exception si cela arrive et de la gérer au niveau de la méthode main.

15. Mettez à jour le main afin de tester ces nouvelles fonctionnalités.