

TRAVAIL PRATIQUE 2

Parachutisme FinHeureuse : application de point de vente

Contexte : Parachutisme FinHeureuse est un centre de parachutisme situé dans les Hautes-Laurentides. Établi depuis 15 ans, il offre à une clientèle adulte de réaliser des sauts en parachute de manière sécuritaire



Votre travail consiste à créer une application point-de-vente. Une application point-de-vente est un système permettant d'enregistrer les entrées, de gérer les paiements ainsi que l'inventaire des clients. Elle est donc installée à la caisse du centre d'accueil des futurs parachutistes.

La première partie de ce travail serait donc la création d'un modèle objet adéquat. Heureusement pour vous, cette étape est déjà réalisée (voir projet dans le répertoire Données:/B43 /TP2). Cependant, pour m'assurer que tous maîtrisent bien le modèle et ses constituants, je vous demanderai de :

- **commenter l'ensemble des classes et des méthodes.**
- **faire un peu de ménage dans l'indentation, remplacer des valeurs par des constantes**

Vous pouvez vous aider des points suivants :

1- Qu'est-ce qu'une Hashtable (Table de Hachage) ?

voir la fichier théorie sur les `Hashtables` sur mon site web

3- fichier `EnsembleClients`

Cette classe, composée également d'une `Hashtable`, représente l'ensemble de tous les clients des magasins `Bradlees` ; ils doivent avoir une `carteMembre` pour réaliser un saut, elle est gratuite et permet de fidéliser la clientèle en permettant aux clients d'accumuler des points bonis. Les clients peuvent également accumuler des points bonis dans d'autres commerces du secteur.

La table de hachage `listeClients` est donc composée de doublets clé-élément (la clé est le numéro de la carte boni du `Client` et l'élément est l'objet `Client` au complet). Toutes ces méthodes sont statiques car cette liste de clients est unique.

5- fichier Commande

Les objets `Commande` représenteront les commandes que les clients présentent au caissier/ère. Ils sont donc entre autres constituées d'un vecteurs de Billets (un `Client` peut faire de 1 à 3 sauts par jour) et comprennent des méthodes afin de calculer les totaux partiels et les taxes.

6- fichier Billet

Un objet `Billet` contient le type de client et le prix qui lui est rattaché. Il n'y a que deux types de clients (réguliers ou aînés-65 ans et plus). Un `Client` peut acheter de 1 à 4 billets pour la journée; chaque billet lui donnant droit à un saut. Les billets ne sont pas remboursables.

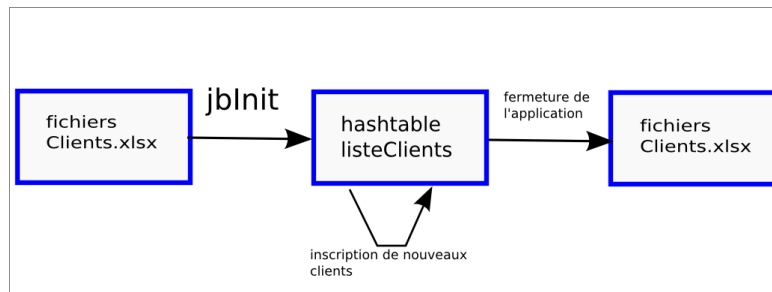
7-fichier Client

Représente les `Clients` qui seront contenus dans la `Hashtable` de l'`EnsembleClient`.

8- Persistance des informations (fichiers Clients.xlsx)

La classe `EnsembleClients` renferme une hashtable qui contenant tous les clients du Centre de parachutisme. Pour faciliter le travail de gestion, les fichiers contenant l'ensemble des clients sont des fichiers Excel (.xlsx); de cette manière, le gérant pourra modifier les adresses ou ajouter des points bonis à des clients directement dans un fichier Excel qu'il peut facilement utiliser.

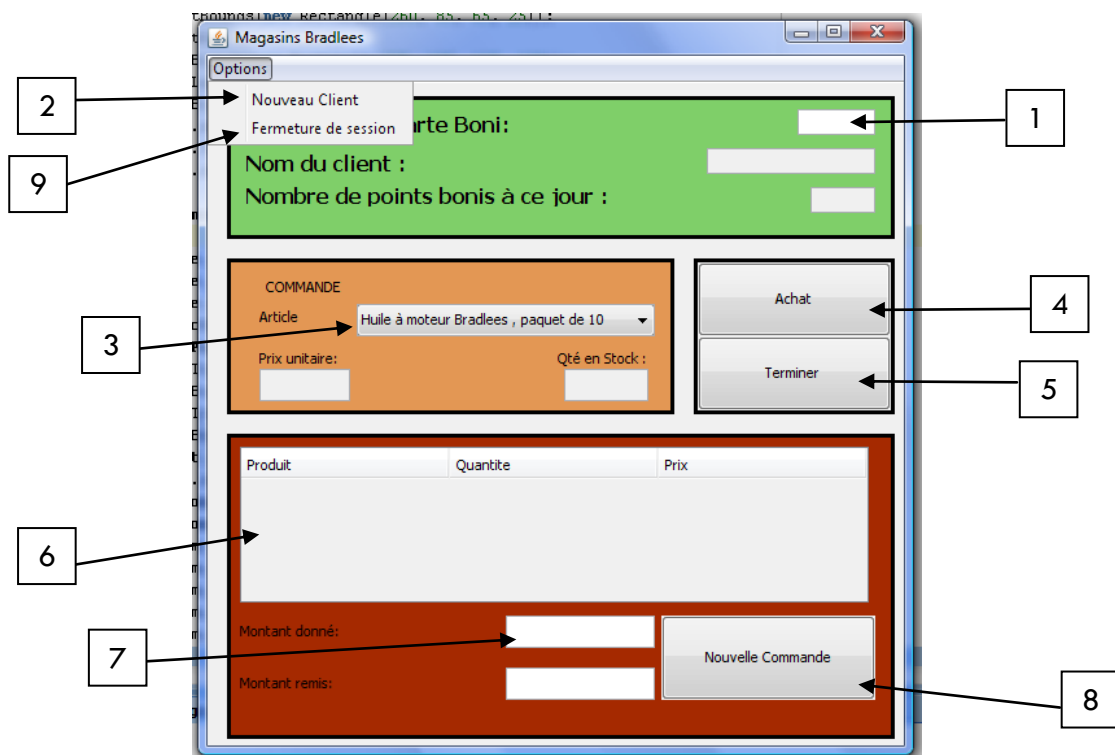
Donc,



La lecture / écriture des fichiers Excel dans Java se fait à l'aide de bibliothèques externes téléchargées (dossier poi 3.9). Ces bibliothèques ont été téléchargées d'un projet Apache : <http://poi.apache.org/index.html>

Vous devez donc faire en sorte de charger les clients contenus dans le fichier Excel dans la table de hachage pour l'utiliser durant le fonctionnement du programme. Aucun accès au fichier Excel ne doit être fait durant le fonctionnement de l'application : uniquement au début et à la fin du fonctionnement de celle-ci.

LE GUI



Vous devez maintenant créer un GUI qui travaillera en relation avec le modèle créé. Vous avez pleine liberté pour les choix de composantes mais les points suivants doivent être présents:

- 1- Lorsqu'un client se présente à la caisse, l'employé entre le code de la carte de membre du client. Avec enter, on voit apparaître les coordonnées du client (son nom et son nombre de points bonis) sur l'interface. Si le numéro entré n'existe pas dans la table de hachage, envoyer un message à l'aide d'un `JOptionPane` et le commis essaie un nouveau numéro de carte.
- 2- Si jamais le client n'a pas de carte de membre, l'employé clique sur un item dans le menu (`JMenuBar`, `JMenu`, `JMenuItem`) en haut de l'interface. Un `JDialog` semblable à celui-ci devra apparaître :

- le numéro de la carte du nouveau client doit être généré aléatoirement (un numéro de 6 chiffres). Vous devez bien sûr vérifier que le numéro généré n'est pas déjà utilisé par un autre client (`Math.random()` ou classe `Random`)
 - Si un nom de client est entré, un clic sur le bouton OK ajoutera un nouveau client à la table de hachage de la classe `EnsembleClients`. Ce client sera éventuellement ajouté au fichier Excel `Clients.xlsx` lorsque la session du préposé sera terminée. Il faut s'assurer que le numéro de membre n'est pas déjà présent dans la table de hachage.
 - Transférez également les informations du nouveau client dans les champs appropriés de la fenêtre principale (voir #1)
- 3- De retour dans l'interface principale, faites en sorte que le caissier puisse saisir le nombre de billets achetés par le membre afin de constituer, peu à peu, l'objet `Commande` liée à la transaction.
 - 4- Lorsque l'employé clique sur *Terminer*, il est temps de calculer les taxes et le total de la facture **à partir de votre modèle** et de les afficher dans la composante `JTable`. Faites en sorte que l'usager puisse choisir d'utiliser une partie de ses points bonis pour payer une partie de sa facture.
 - 5- On doit pouvoir saisir le montant donné par le client (paiement comptant seulement) . S'il est supérieur au total de la facture, on affiche le montant à remettre dans le champ approprié et on affiche le nouveau total de points bonis. Sinon, on fait apparaître un `JOptionPane` mentionnant le problème au caissier.

- 6- La transaction est conclue, on efface les différents champs et on accueille le prochain client avec le sourire 😊
- 7- Quand le caissier a terminé son quart de travail, il sélectionne l'item Fermeture de session dans le menu du haut de l'interface. On doit sauvegarder les nouvelles données dans le fichier Excel existant :
 - les nouveaux clients enregistrés durant le quart de travail
 - le nouveau nombre de points des clients existantsEn pratique, il m'apparaît plus simple de réécrire l'ensemble des données dans les fichiers

BONUS :

- Générez un nouveau fichier Excel contenant le montant des transactions de chaque client pour une session de travail donnée
- Ajoutez un bouton « annuler la transaction » permettant d'annuler une commande en cours de route. Pensez à l'impact sur les quantités en stock, par exemple

N'OUBLIEZ PAS :

- Ce document ne peut pas tenir compte de toutes les situations possibles. Servez-vous de votre GBS et améliorez au maximum l'ergonomie de votre interface.