2020-2021

Mini-Projet : Algorithmique et programmation en python

Pr. Imane DAOUDI

Gestion d'une pharmacie

Écrivez un programme orienté objets (en Python ou Java) qui permet de gérer une pharmacie.
La pharmacie gère des clients et des médicaments. Un client est caractérisé par un **nom** et un **crédit**. Le crédit représente la somme que ce client doit à la pharmacie. Le crédit peut être négatif si le client a versé plus d'argent que le montant. Un médicament est caractérisé par un **nom**, un **prix** et un **stock**. On souhaite avoir une application qui propose les options suivantes :

- **affichage(..)** permet d'afficher les clients et leurs crédits respectifs ainsi que les médicaments et leurs stocks respectifs.

- **approvisionnement(..)** permet d'approvisionner le stock d'un médicament. Le nom du médicament à approvisionner ainsi que la quantité à ajouter au stock doivent être lus depuis le terminal (le clavier). Lorsque le nom du médicament est introduit, il faut vérifier qu'il s'agit bien d'un nom connu dans la liste des médicaments de la pharmacie. Si le médicament se trouve dans la liste alors on incrémente son stock.
- **enregistrement(...)** permet d'ajouter un nouveau médicament dans la liste des médicaments de la pharmacie

- **achat(..)** permet de traiter un achat fait par un client. L'achat porte sur un médicament donné dans une quantité donnée. Pour cette transaction le client paie un certain prix. Une opération d'achat aura pour effet de déduire la quantité achetée du stock du médicaments correspondant et d'augmenter le crédit du client (d'un montant équivalent au montant de l'achat moins la somme payée).
  Les noms du client et du médicament doivent être lus depuis le terminal (le clavier). Le programme doit boucler jusqu'à introduction de noms connus aussi bien pour les clients que les médicaments. Ces procédures de vérification seront prises en charge par les méthodes lireClient et lireMedicament (voir plus bas). La quantité achetée et le montant payé sont aussi lus depuis le terminal (le clavier). Ils seront supposés corrects.

- **quitter(..)** affiche le message ``programme terminé!''.

Vous définirez une méthode auxiliaire lireClient(..) prenant comme paramètre un tableau de clients. Elle permettra de lire le nom d'un client depuis le terminal et de vérifier si ce client existe dans le tableau des clients. Dans ce cas le client sera retourné. Cette méthode doit boucler jusqu'à ce qu'un client soit trouvé. Elle sera utilisée par la méthode achat(..).
Une méthode similaire, lireMedicament(..) sera fournie pour les médicaments. Elle sera utilisée par les méthodes achat(..) et approvisionnement(..).

Vous êtes libre de définir, en plus de ces méthodes, toutes celles que vous jugerez nécessaires.
Proposer un jeu de tester pour vérifier le bon fonctionnement de votre application