BTS SERVICES INFORMATIQUES AUX ORGANISATIONS

POO / JAVA • TP PHARMACIE

I. CONDITIONS DE RÉALISATION

- Java SE Development Kit
- EDI (Visual Studio Code)

II. TRAVAIL À RÉALISER

Écrire les classes nécessaires à la gestion d'une pharmacie dont la classe principale est la suivante :

```
import java.util.Scanner;

/**
 * Classe Pharmacie
 */
class Pharmacie {

    // La classe Scanner est utilisée pour réaliser une entrée au clavier
    private static Scanner scanner = new Scanner(System.in);

    public static void main (String args[]) {

        Client[]  clients = new Client[2];
        Medicament[]  medicaments = new Medicament[2];

        // Le constructeur prend 2 paramètres, le nom du client et son solde
        clients[0] = new Client("Karamazov",0.0);
        clients[1] = new Client("Deray",0.0);

        // Le constructeur prend 3 paramètres, le nom du médicament, son prix
        // et son stock
        medicaments[0] = new Medicament("Transipeg", 20.40, 5);
        medicaments[1] = new Medicament("Effortil",19.15, 5);

        int choix;

        do {
            choix = menu();

            switch (choix) {
            case 1 :
```

```
                achat(clients, medicaments);
                break;
            case 2 :
                approvisionnement(medicaments);
                break;
            case 3 :
                affichage(clients, medicaments);
                break;
            case 4:
                quitter();
            }
        }
        while (choix < 4);
    }
    // Les méthodes utilitaires

    static int menu() {
        int choix = 0;
        System.out.println();
        System.out.println();
        System.out.println("1 : Achat de médicament");
        System.out.println("2 : Approvisionnement en  médicaments");
        System.out.println("3 : Etats des stocks et des crédits");
        System.out.println("4 : Quitter");
        while ((choix != 1) && (choix != 2) && (choix != 3) && (choix != 4)) {
            choix = scanner.nextInt();
        }
        return choix;
    }
}
```

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 (de type String), un prix (de type double) et un stock (de type int). Les méthodes à compléter auront les caractéristiques suivantes :

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

- `approvisionner()` 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. 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. Le programme doit boucler jusqu'à introduction d'un nom correct. Cette procédure de vérification sera prise en charge par la méthode `lireMedicament()` décrite plus bas.

- `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 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. 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.

Votre programme doit être bien modularisé à la fois sous forme de méthodes auxiliaires de la méthode `main()` et sous forme de classes pour les objets du programme. Chaque variable et méthode doit être déclarée dans la classe la plus adaptée.

Dans chaque classe liée à un objet, il faut utiliser le mot-clé `private` autant que possible. En particulier, toutes les variables d'instances seront privées.

Vous pouvez utiliser les méthodes `scanner.nextInt()`, `scanner.nextDouble()` et `scanner.nextLine()` (respectivement pour la saisie d'un entier, d'un nombre décimal et d'une chaine de caractères).