TD1

Gestion des caisses d'un supermarché

Nous voulons simuler la gestion du flux des clients d'un supermarché aux caisses. Pour cela, nous pouvons prendre de la réalité quelques conditions :

- Un client est affecté à au plus une caisse ;
- Plusieurs clients peuvent être affectés à une seule caisse ;
- Si on modélise une caisse par une classe d'objets CAISSE, elle aura un attribut « ouverte » qui permet de définir l'état de la caisse aux clients (0 : fermée, 1 : ouverte) ;
- Si on modélise un client par une classe d'objets CLIENT, elle aura un attribut « duree_prevue » qui donne la durée prévue pour le traitement du caddy du client. Le temps d'attente d'un nouveau client arrivant à la caisse est la somme des durées prévues de traitement de tous les clients qui sont devant lui ;
- La classe CLIENT aura aussi un attribut « rang » qui donne le rang que le client occupe à la caisse. Si un client à un rang=4 signifie qu'il y a 3 clients devant lui.

Pour développer ce simulateur de caisses nous pouvons suivre les étapes suivantes :

1. Ecrire la classe CAISSE avec les attributs privés id, ouverte et la liste clients ;
2. Ecrire le constructeur de la classe CAISSE qui prend en argument le numéro de la caisse et un booléen pour définir si la caisse est ouverte ou fermée ;
3. Ce constructeur initialise les variables d'instance de la classe ;
4. Créer la classe CLIENT avec les attributs privés : rang et duree_prevue ;
5. Ecrire le constructeur de la classe CLIENT qui initialise son rang à 0 et initialise sa durée prévue à 0 aussi ;
6. Ecrire la méthode setDureePrevue() qui prend en argument une durée pour mettre à jour la durée prévue de traitement du chariot du client. Cette durée dépend du contenu du chariot ;
7. Ecrire la méthode getDureePrevue() qui lit l'attribut duree_prevue ;

8. Ecrire les deux méthodes setRang() et getRang() pour mettre à jour et lire le contenu de l'attribut « rang » ;

9. Ecrire les méthodes suivantes de la classe CAISSE : ouvrir, fermer, isOuverte, passerClient, ajoutClient et nbClients tel que :

ouvrir : permet d'ouvrir la caisse ;
fermer : permet de fermer la caisse ;
isOuverte : permet de vérifier si la caisse est ouverte ;
passerClient : permet de faire passer un client traité. Cette fonction enlève le client de rang 1 de la file d'attente de la caisse et décale les rangs des clients qui restent dans la file. Si la caisse n'a pas de clients en attente, alors cette fonction ne fait rien ;
ajoutClient : permet d'ajouter un client à la liste des clients de la caisse en mettant à jour son rang, et en initialisant sa durée prévue de traitement par un nombre aléatoire simulant la durée de traitement de son chariot ;
nbClients : donne le nombre de clients à la caisse. Si la caisse est fermée, cette fonction renvoie 0.

10. Ecrire la méthode temps_traitement() de la classe CAISSE qui renvoie la durée de traitement de tous les clients de cette caisse ou bien 0 si n'y a pas de clients à la caisse ;

11. Créer la classe MAGASIN la classe principale ;

12. Ecrire la fonction creeCaisses() qui permet de créer 10 caisses dans la base données de ce magasin qui seront fermées au départ ;

13. Ecrire les deux fonctions arriveeClient() et sortieClient() qui permettent de créer un client à son entrée au magasin et de l'enregistrer dans la base et de le supprimer de la base à son sortie ;

14. Ecrire la fonction meilleureCaisse() qui permet de retourner la caisse dont la durée d'attente est minimale ;

15. Ecrire la fonction affectationClient() qui prend en argument un client pour l'affecter à la meilleure caisse ;

16. Dans la fonction main() de cette classe, écrivez le scénario suivant :

    a. Création des caisses et enregistrement des caisses ;

    b. Ouverture des caisses ;

    c. Arrivée et enregistrement des clients ;

    d. Affectation des clients aux caisses ;

    e. Sortie des clients ;