

Travail 2

Remise : lundi 2 mars avant minuit via l'option remise de Léa

enlevez les 2 dossiers x64 et le dossier caché .vs

Pondération : 10%

But :

- Mettre en pratique la relation de composition entre classes
- Continuer à pratiquer les notions de programmation objet et la relation d'association

Le travail peut s'effectuer en équipe de 2 personnes maximum.

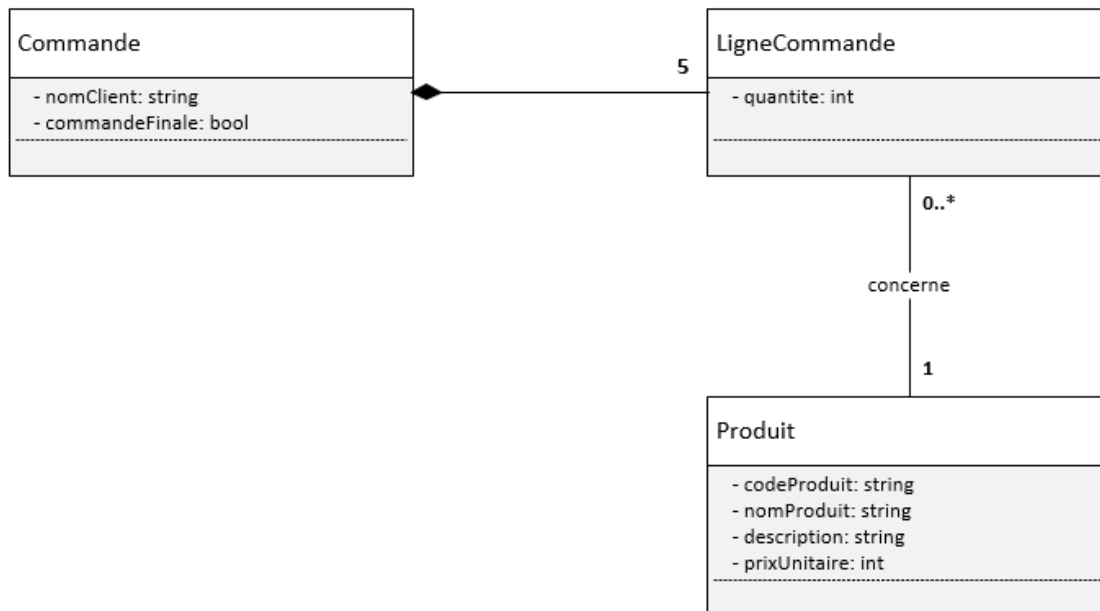
Description :

Le travail consiste à aider les célèbres frères Weasley, que tout le monde connaît, à gérer leurs commandes pour leur magasin « Farces pour sorciers facétieux ». Le magasin offre différents produits que les clients peuvent acheter en plusieurs exemplaires s'ils le désirent.

Schéma de classes

Le schéma de classes suivant sera utilisé afin de représenter les classes à manipuler et leurs liens.

- La classe Commande représente tout simplement une commande d'un client, laquelle comporte jusqu'à 5 produits commandés en une certaine quantité.
- La classe Produit représente évidemment un produit à vendre.



Transformez ce schéma afin d'obtenir le schéma des classes à programmer. Faites-moi approuver votre schéma.

Exigences générales

- Le programme doit être fait à l'aide de Visual Studio C++ en mode console.
- Faites attention pour respecter les principes de la programmation objet.
- Respectez aussi les bonnes techniques de programmation. Portez une attention particulière aux noms utilisés ainsi qu'au découpage du programme en fonctions.
- Utilisez les constantes indiquant le nombre d'éléments maximum dans les tableaux.
- Continuez à utiliser git en faisant des commits au fur et à mesure de l'avancement du travail ainsi que quelques push sur gitHub pour avoir des copies.

Programmation des classes

Complétez la programmation des classes tout en respectant le schéma que vous m'avez fait approuver.

- Vous ne pouvez pas modifier la classe Produit car vous n'avez pas le fichier .cpp. Vous avez seulement le fichier .h et le fichier .obj (compilation du .cpp).
- Chaque classe doit être programmée à l'intérieur de ses propres fichiers (.h et .cpp)
- Chaque classe doit **posséder un constructeur par défaut et un constructeur par recopie.**
- **Assurez-vous de dédier les responsabilités aux bonnes classes.**
- **Assurez-vous de respecter les façons de faire entourant les classes en relation de composition.**
- **Assurez-vous de respecter la relation d'association.**

Programmation de l'application

Vous possédez déjà une partie du programme principal.

- Les produits à vendre sont créés.
- L'affichage du menu et la fonction main() sont déjà faits afin de contrôler le déroulement du programme.
- **Complétez CreerCommande()** : pour l'ajout d'une nouvelle commande
 - Vous devez obligatoirement utiliser l'objet 'laCommande' pour procéder à l'ajout de la commande.
 - Elle doit évidemment demander le nom du client.
 - Il doit être possible d'ajouter plusieurs produits à la commande en demandant le code du produit à commander ainsi que la quantité désirée. Si le code du produit entré n'existe pas vous devez afficher un message.
 - Une fois que vous avez terminé de demander les informations de la commande à l'utilisateur, vous devez obligatoirement appeler la fonction AjouterCommande(). Cela permettra de vérifier le fonctionnement du constructeur par recopie. La commande doit être ajoutée dans le 1^{er} emplacement vide du tableau.

- Afficher les commandes
 - Pour chaque commande, faites afficher :
 - le nom du client
 - tous les produits commandés : code et nom du produit, prix unitaire, quantité commandée et le total pour ce produit
 - le grand total de la commande.
- 😊 Assurez-vous de tester tous les cas possibles : une commande comportant 1 seul produit en 1 exemplaire, commande comportant 1 seul produit en N exemplaires, ...
- Finaliser les commandes
 - Pour chaque commande **non finale** :
 - Faites afficher la commande : mêmes informations que pour l’affichage ci-haut.
 - Demandez à l’usager s’il désire finaliser cette commande.
 - Si oui, demandez s’il désire modifier des quantités.
 - Si oui, demandez le code du produit ainsi que la nouvelle quantité et procédez à la modification de la quantité pour le produit indiqué.
 - Affichez un message indiquant que l’usager s’est trompé s’il entre un code non présent sur cette commande.
 - Affichez un message indiquant que la modification s’est déroulée avec succès si le code entré est valide pour cette commande.
 - Il doit être possible de procéder à plusieurs modifications pour une même commande.
 - Une fois les modifications de quantité effectuées, indiquez que la commande est maintenant finale.

Critères de correction et répartition des points :

- Respect du schéma
- Application des règles de la programmation objet : encapsulation, relation, ...
- Responsabilités aux bonnes classes
- Respect des exigences demandées
- Respect du standard et des bonnes pratiques de programmation
- Utilisation adéquate des objets
- Fonctionnement
- Interface

Répartition des points

- | | |
|--|---------------|
| • Classes Commande et LigneCommande | 20 pts |
| • Ajout d’une commande | 20 pts |
| • Affichage des informations des commandes | 20 pts |
| • Calcul et affichage des totaux demandés | 15 pts |
| • Modification des commandes | 25pts |