INF1010

Programmation Orientée-Objet

Travail pratique #5

Fonctions, classes génériques et librairies STL

Objectifs :	Permettre à l'étudiant de se familiariser avec les concepts de foncteur, de fonction et de classe générique et avec les librairies standard.
Remise du travail :	mardi 7 Avril avant, 8h
Références :	Notes de cours sur Moodle & Chapitres 11 à 14, 16 et 20 du livre Big C++ 2e éd.
Documents à remettre :	Les fichiers .cpp et .h complétés réunis sous la forme d'une archive au format .zip.
Directives :	Directives de remise des Travaux pratiques sur Moodle Les en-têtes (fichiers, fonctions) et les commentaires sont obligatoires. Veuillez suivre le guide de codage

Travail à réaliser

Dans le travail pratique #2, vous avez réalisé un programme qui permet une gestion simple de différents paniers d'articles.

Dans ce travail pratique #5, nous voulons réaliser un système pour la gestion d'une plateforme de commerce. En plus de gérer les paniers comme votre programme précédent, il faudra également prendre en considération la gestion des clients.

Afin d'implémenter votre programme, vous profiterez des librairies standard du C++. Il vous est donc interdit d'utiliser des boucles simples (ex : for (int i=0; i<n; i++)) toutes les classes et le programme principal Vous pouvez cependant utiliser des itérateurs pour parcourir vos conteneurs.
N'hésitez pas à utiliser les algorithmes STL (for_each, sort, find, count, etc.) couplés avec des foncteurs.

Vous devez implémenter les classes Panier, Article, PanierArticles, Client et Commerce.

Classe Générique Panier

Cette classe ne devrait pas prendre plus d'une séance de TP à être implémentée.

La classe Panier est une classe générique permettant de contenir plusieurs objets de type quelconque (qu'on appellera ici type T).

La classe comprend les attributs suivants :

- Un identifiant unique (id_) de type entier non signé.
- Une liste de pointeurs d'instances de T (liste_). Utilisez le conteneur STL List.

Les méthodes suivantes doivent êtres implémentées :

- Un constructeur par paramètre prenant comme paramètre l'identifiant.
- Des accesseurs pour l'ID et le conteneur d'éléments du panier.
- Une méthode *ajouter* avec un paramètre de type T. La méthode ajoute le paramètre à la liste du panier.
- Une méthode *obtenirPlusPetitElement* qui retourne l'élément le plus petit de la liste du panier. Utiliser la fonction *min_element*. (selon l'opérateur < du typeName T)
- Une méthode *obtenirPlusGrandElement* qui retourne l'élément le plus grand de la liste du panier. Utiliser la fonction *max_element*.
- Une méthode *supprimer* avec un identifiant (entié non signé) en paramètre. Cette méthode supprime un seul élément de la liste du panier. Vous devez utiliser la fonction *find_if* de la STL avec la méthode *erase* du conteneur (attribut).
- Une méthode *supprimer* qui prend un prédicat unaire en paramètre. Ce prédicat est un template de la méthode. Cette méthode supprime tous les éléments du panier qui ne respecte pas ce prédicat.
- Une surcharge de *operator*<<. Cette méthode affiche tous les éléments du panier. **Aucune boucle** n'est autorisée pour l'implémentation de cette méthode.

Classe Client

Il s'agit d'une classe simple avec les attributs suivants :

- Un identifiant (id_) de type entier non signé.
- Un nom (nom_) de type string.
- Un prénom (prenom_) de type string.
- Le solde (solde_) de type float.

Les méthodes suivantes doivent êtres implémentées :

- Un constructeur par paramètres qui initialise tous les attributs.
- Des accesseurs pour tous les attributs.
- Surcharge de *operator*< . Le seul attribut considéré pour la comparaison sera le nom.
- Une surcharge de *operator*<<. Cette méthode affiche sur une ligne les informations du client.

- Une méthode qui diminue le solde du client selon le paramètre de la méthode.

Classe Article

Il s'agit d'une classe simple avec les attributs suivants :

- Un identifiant (id_) de type entier non signé.
- Un nom (nom_) de type string.
- Un prix (prix_) de type float.

Les méthodes suivantes doivent êtres implémentées :

- Un constructeur par paramètres qui initialise tous les attributs.
- Des accesseurs pour tous les attributs.
- Surcharge de *operator* < qui compare les prix des articles.
- Surcharge de *operator*<< qui permet d'afficher un panier.

Classe PanierArticles

Cette classe est une classe dérivée de la classe générique Panier dont le typeName est la classe Articles.

Les méthodes suivantes sont implémentées :

- La méthode *sommeArticles* qui retourne la somme des prix des articles dans le panier.
- La méthode *obtenirMoyenne* qui retourne la moyenne des prix des articles.
- Une méthode *trier* qui retourne une liste d'éléments du panier triée. Il ne faut pas modifier la liste d'élément du panier courant; il faut retourner une nouvelle liste. Cette méthode retourne une liste d'article en ordre décroissant du prix.
- Surcharge de *operator*<< qui permet d'afficher un panier.

Classe Commerce

La classe Commerce s'occupe de faire le lien entre un client et sa commande (panier).

La classe comprend les attributs suivants :

- Une map dont les clés sont des clients et un pointeur de paniers d'articles. Nommez l'attribut « mapClientPanier_ ». La clef de la map est l'Id du client.

Les méthodes suivantes doivent être implémentées :

- La méthode ajouterCommande() qui prend en paramètres un client et un pointeur de panier d'articles. La méthode ajoute le client ainsi que son panier à la map. La méthode renvoie vraie si le client a été ajouté et faux s'il était déjà présent et donc pas ajouté.
- La méthode ajouterArticle() qui prend en paramètres un client et un pointeur d'article. La méthode ajoute l'article au panier du client. La méthode renvoie vraie si l'article a bien été ajouté et faux s'il ne l'a pas été car le client n'est pas présent dans la map.

- La méthode supprimerCommande() qui prend en paramètre un client. La méthode supprime ce client de la map. La méthode renvoie vraie si le client a bien été supprimé et faux si le client n'est pas présent.
- La méthode supprimerArticleCommande() qui prend en paramètres un client et un pointeur d'article. La méthode retire l'article du panier du client en question. La méthode renvoie vraie si l'article a été retiré et faux si le client n'est pas présent.
- La méthode appliquerRabais() qui prend en paramètres un client et un foncteur. Cette méthode doit appliquer un rabais d'un certain pourcentage sur l'ensemble du panier du client visé. Le pourcentage est un paramètre ou attribut du foncteur.
- La méthode payerPanier qui prend en paramètre un client et un montant d'argent et déduit le solde du client.
- La méthode afficher(Client client) qui affiche le panier d'un client.
- La méthode afficher(string nomClient) qui affiche le panier d'un client selon son nom.
- La méthode afficherParOrdreAlphabetique() qui affiche les paniers selon l'ordre alphabétique des clients. **Rappel :** Les éléments d'une map sont classés selon leurs clés.
- La méthode afficherParPrixMoyenDecroissant() qui affiche les paniers classés en ordre décroissant selon le prix des articles dans le panier.

Foncteur

Écrire un foncteur pour un algorithme de la librairie STL qui permet de trouver les clients qui ont payé au moins un certain pourcentage de leur solde. Le pourcentage est un attribut.

Main.cpp

Le programme principal contient des directives à suivre pour instancier différents objets et essayer les différentes méthodes implémentées.

Correction

La correction du TP5 se fera sur 20 points. Voici les détails de la correction:

- (10 points) Compilation et exécution exacte des différentes méthodes ;
- (04 points) Utilisation adéquate des libraires STL ;
- (04 points) Implémentation correcte des classes génériques ;
- (01 point) Respect des consignes de l'énoncé ;
- (01 point) Documentation du code et norme de codage ;