

INF1010
Programmation Orientée-Objet
Hiver 2019

Travail pratique #4
Polymorphisme, Héritage multiple

Objectifs :	Permettre à l'étudiant de se familiariser avec le polymorphisme, la conversion dynamique de type, l'héritage multiple, les fonctions virtuelles, les classes abstraites et les interfaces.
Remise du travail :	Lundi 25 mars 2019, 8h (AM)
Références :	Notes de cours sur Moodle & Chapitre 19 du livre Big C++ 2e éd.
Documents à remettre :	Tous les fichiers .cpp et .h exclusivement complétés et réunis sous la forme d'une archive au format .zip.
Directives :	<u>Directives de remise des Travaux pratiques sur Moodle</u> Les en-têtes (fichiers, fonctions) et les commentaires sont obligatoires. Les travaux dirigés s'effectuent obligatoirement en équipe de deux personnes faisant partie du même groupe. Pas de remise possible sans être dans un groupe. <u>Veuillez suivre le guide de codage</u>

Travail à réaliser

Le travail consiste à continuer l'application PolyFood commencée aux TP précédents en y intégrant le polymorphisme, la conversion dynamique de type, l'héritage multiple, les fonctions virtuelles, les classes abstraites et l'interface.

Les fonctions virtuelles permettent à une classe dérivée de surcharger une méthode et que celle-ci soit appelée par le compilateur C++ sur un pointeur de cette classe, même si le pointeur a été défini comme un pointeur de la classe mère. De plus, elles peuvent servir à définir des classes abstraites en terminant la définition d'une fonction virtuelle par un `= 0`. Une classe ayant au moins une méthode abstraite est une classe abstraite et on ne peut pas créer d'objet de cette classe.

Une classe possédant seulement que des fonctions virtuelles pures ne peut être instanciée et est appelée une interface.

Pour vous aider, les fichiers du TP précédent vous sont fournis. Vous n'avez qu'à implémenter les nouvelles méthodes ou les méthodes à modifier décrites plus bas. Les attributs ou méthodes qui ne sont plus nécessaires ont été supprimés.

Plusieurs modifications et améliorations ont été apportées par rapport au TP3.

La structure du fichier polyfood.txt a changé, et les méthodes de lecture ont déjà été modifiées en conséquence.

La classe LectureFichierEnSections permet de lire le fichier en section. Si vous exécutez vos programmes sous mac, il faut modifier une seule ligne de la méthode lireLigne(). Lire le commentaire.

Le fichier def.h contient toutes les constantes et les énumérations des classes.

Le fichier initDebogageMemoire() permet d'avoir un rapport des fuites de mémoire dans la fenêtre de "sortie" de Visual Studio. Mais il ne fonctionne pas sous Mac.

Vous devez ajouter des destructeurs,

Il faut utiliser le mot clé `const` chaque fois que cela vous semble pertinent.

L'affichage du programme doit être aussi proche que possible de celui présenté en annexe.

Vous ne devez pas modifier les signatures des méthodes fournies.

Il n'est pas indiqué si les méthodes sont virtuelles ou virtuelles pures. Pour toutes les classes, il faut identifier les méthodes qui doivent être virtuelles ou virtuelles pures.

Il faut utiliser les fichiers fournis, plutôt que de continuer avec vos fichiers du TP2.

Écrire les destructeurs lorsque c'est nécessaire.

Pour plus de précision sur le travail à faire et les changements à effectuer, veuillez-vous référer aux fichiers .h et à l'énoncé.

Afin de diversifier leurs offres, le restaurant PolyFood ajoute à son menu des différents types de plats avec l'introduction de plat biologique et la possibilité de personnaliser un certain nombre d'ingrédients dans un plat.

Les nouvelles classes de plats dérivent de la classe Plat : PlatBio, PlatVege, PlatBioVege. La classe PlatCustom a été supprimée.

Un programme de fidélité est également créé pour les clients pour créer des nouvelles classes de clients qui dérivent de la classe Client : ils sont donc répartis en trois catégories qui donnent accès à différents avantages : **les clients occasionnels, les clients réguliers et les clients prestigieux**. Les

clients seront désormais représentés par des objets plutôt que par des nombres. Un programme de livraison est aussi disponible dans le restaurant et accessible une catégorie restreinte de client.

Pour vous aider, les fichiers corrigés du TP précédent vous sont fournis. Vous n'avez qu'à implémenter les nouvelles méthodes décrites plus bas et les méthodes à modifier vous ont été indiquées.

Classe Vege

La classe Vege est une classe abstraite et il est impossible de créer une instance de cette classe. Elle permet de conserver les attributs protéines, vitamines et minéraux. Il s'agit donc de déterminer les méthodes virtuelles ou vituelles pures. Il faut implémenter la méthode afficherVege().

Classe Plat

La classe Plat est un plat régulier comme dans le tp3.

Les méthodes suivantes doivent implémentées :

- La méthode afficherPlat() affiche les attributs du plat (voir l'affichage à la fin de l'énoncé) : nom, le prix et le coût comme présenté dans l'annexe.
- La méthode clone() retourne un objet alloué dynamiquement qui est une copie de l'objet courant.

Classe PlatBio

Il s'agit d'un plat biologique qui est ajouté au menu du restaurant et qui dérive de la classe de base Plat.

L'attribut ecotaxe_est maintenant protected.

Les méthodes suivantes doivent implémentées:

- Le constructeur par paramètres.
- La fonction afficherPlat() qui affiche les attributs du platBio (voir l'affichage à la fin de l'énoncé)
- La méthode getPrixDeRevient() retourne simplement le prix - le coût + l'ecotaxe.
- La méthode clone() retourne un objet alloué dynamiquement qui est une copie de l'objet courant.

Classe PlatBioVege

PlatBioVege est une classe qui dérive de PlatBio et de Vege. Un plat végétarien a un apport nutritif qui est plus important que les autres. Encore plus encore quand il s'agit un plat biologique, puisqu'il n'y a pas de pesticides. Elle a doit définir la méthode calculerApportNutritif.

Les méthodes suivantes doivent être implémentées:

- Le constructeur par paramètres.
- La méthode afficherPlat qui affiche les attributs.
- La méthode calculerApportNutritif doit retourner l'apport nutritif qui se calcule comme suit : $(vitamines_ * proteines_ / mineraux_) * RAPPORT_NUTTRITIF_BIO * AVANTAGE_SANS_PESTICIDE$
- La méthode clone() retourne un objet alloué dynamiquement qui est une copie de l'objet courant.

Classe PlatVege

Il s'agit d'un plat végétarien qui est ajouté au menu du restaurant et qui dérive des classes de base Plat et de Vege. Un plat végétarien a un apport nutritif plus important que les autres.

Les méthodes suivantes doivent être implémentés:

- Le constructeur par paramètres
- La méthode afficherPlat qui affiche les caractéristiques du plat.
- La méthode calculerApportNutritif retourne l'apport nutritif qui se calcule comme suit : $(vitamines_ * proteines_ / mineraux_) * RAPPORT_NUTTRITIF$
- La méthode clone() retourne un objet alloué dynamiquement qui est une copie de l'objet courant.

Classe Client

La classe de base est maintenant une **classe abstraite**! Il est impossible de créer d'instance de cette classe.

Les méthodes suivantes doivent être virtuelles pures :

getNbPoints() : il n'y a aucune implémentation de cette méthode dans la classe Client.

afficherClient() : écrire les instructions qui affiche le prénom, le nom du client et la table si le pointeur de Table est différent de nullptr ;

getReduction() : il n'y a aucune implémentation de cette méthode dans la classe Client.

Classe ClientOccasionnel

Cette nouvelle classe dérive de la classe Client. Elle représente les clients occasionnels du restaurant qui ne participent pas au programme de fidélité. Ils ne bénéficient d'aucune livraison et d'aucune réduction.

Les méthodes suivantes doivent être implémentées :

- Le constructeur par paramètres.
- La méthode afficherClient qui affiche les attributs.
- La méthode getNbPoints. Un client Occasionnel n'a aucun point.
- La méthode getReduction. Le client Occasionnel n'a aucune réduction.

Classe ClientRegulier

Elle représente les clients réguliers du restaurant qui participent au programme de fidélité et dérive de la classe Client. Les clients ne sont pas admissibles aux livraisons, mais bénéficient d'un taux de réduction TAUX_REDUC_REGULIER à partir d'un certain nombre de points.

Elle contient l'attribut protégé suivant :

- nbPoints_ : nombre de points de fidélité.

Les méthodes suivantes doivent être implémentées:

- Le constructeur par paramètres
- La méthode afficherClient qui affiche les attributs du client.
- La méthode getReduction() qui retourne la réduction. Si le nombre de points > SEUIL_DEBUT_REDUCTION) alors la réduction = (-montant * TAUX_REDUC_REGULIER)

Classe ClientPrestige

Elle représente les clients les plus assidus du restaurant qui participent au programme de fidélité et dérive de la classe ClientRegulier. Les clients sont admissibles aux livraisons et bénéficient d'un taux de réduction TAUX_REDUC_PRESTIGE. Les plus assidus parmi eux, dont le solde de points dépasse le seuil de livraison gratuite ne paient pas les livraisons.

Elle contient l'attribut privé suivant :

- zoneHabitation_ : la valeur du type énuméré ZoneHabitation et représente la zone d'habitation du client

Les méthodes suivantes doivent être modifiées ou implémentées:

- Le constructeur par paramètres.
- La méthode afficherClient qui affiche les attributs.
- La méthode getReduction qui calcule la réduction en fonction du TAUX_REDUC_PRESTIGE et vérifie si le nombre de points < SEUIL_LIVRAISON_GRATUITE alors les frais de livraison sont ajoutés à la livraison.

Classe Table

On vous donne la définition et l'implémentation de cette classe.

Classe Menu

Dans les attributs, nous utilisons maintenant deux vecteurs, le premier est le même qu'avant, soit un vecteur contenant tous les plats du menu. Le deuxième contient les plats qui sont uniquement végétariens, ce qui va permettre aux clients de trouver rapidement les plats végétariens dans le menu (les objets de PlatBioVege et PlatVege peuvent être référencer par deux pointeurs de base Plat et Vege). Le vecteur listePlatsVege_ n'alloue aucune espace mémoire, il stocke seulement les pointeurs de plats végétariens.

Le fichier à lire aussi a changé, mais la méthode de lecture vous est donnée, donc il n'est pas nécessaire de la modifier.

Les méthodes suivantes doivent être implémentées :

- Le constructeur de copie.
- Le destructeur.
- L'opérateur += reçoit un pointeur de la classe de base Plat. La surcharge doit ajouter le pointeur dans les deux vecteurs. Pensez à utiliser le **dynamic_cast**.
- L'opérateur = qui écrase les attributs du menu par les attributs du menu passé en paramètre et qui renvoie ensuite une référence au menu. Utiliser la méthode allouerPlat().
- La surcharge de l'opérateur << affiche tout le menu et aussi le menu entièrement végétarien.

Classe Restaurant

La définition et l'implémentation de cette classe vous sont données. Il ne faut pas la modifier. Le = delete après une méthode signifie que cette méthode ne pourra être jamais utilisée avec un objet de cette classe.

Classe Taxable

Cette classe est une interface. Il y a seulement deux méthodes non implémentées. Il ne faut pas la modifier.

Classe Boisson

Cette classe doit dériver de la classe Taxable. Il faut modifier la définition de cette classe et écrire l'implémentation. Le pourcentage de la taxe est de 0.12.

Classe PlatVege modifiée

Cette classe doit aussi dériver de la classe Taxable en plus de l'héritage de Vege et Plat.
Il faut modifier la définition de cette classe et écrire l'implémentation des nouvelles méthodes ajoutées. Le pourcentage de la taxe est de 0.07.

Fichier main.cpp

Il ne faut pas modifier le fichier main.cpp.

Affichage

Votre affichage devrait avoir une apparence semblable à celle de ci-dessous.

1er affichage PolyFood :

```
=====Exemple de menu=====
PLAT ----Soupe - 10 $ (5$ pour le restaurant)
Plat Bio   comprend une Taxe ecologique de :1.5$
ET PLAT VEGE Soupe   vitamines 2 Proteines 3 Mineraux 1
PLAT ----Oeuf - 12 $ (4.5$ pour le restaurant)
Plat Bio   comprend une Taxe ecologique de :2.3$
PLAT ----Pain - 5 $ (2$ pour le restaurant)
PLAT VEGE Pain   vitamines 6 Proteines 7 Mineraux 2
(Apport nutritif 16.8mg)
PLAT ----Crepes - 6 $ (2$ pour le restaurant)
PLAT VEGE Crepes   vitamines 9 Proteines 8 Mineraux 1
(Apport nutritif 57.6mg)
PLAT ----Gratin - 23 $ (8$ pour le restaurant)
PLAT ----Pancakes - 7 $ (2$ pour le restaurant)
PLAT ----Homous - 8 $ (3$ pour le restaurant)
Plat Bio   comprend une Taxe ecologique de :0.8$
ET PLAT VEGE Homous   vitamines 1 Proteines 2 Mineraux 3

MENU ENTIEREMENT VEGETARIEN
PLAT VEGE Soupe vitamines 2 Proteines 3 Mineraux 1
PLAT VEGE Pain   vitamines 6 Proteines 7 Mineraux 2
PLAT VEGE Crepes   vitamines 9 Proteines 8 Mineraux 1
PLAT VEGE Homous   vitamines 1 Proteines 2 Mineraux 3
Martin a une place au resto
Amy n'a pas une place au resto
John a une place au resto
Moussa a une place au resto
Andree a une place au resto
Marie a une place au resto
Le restaurant PolyFood n'a pas fait de benefice ou le chiffre n'est pas
encore calcule.
-Voici les tables :
La table numero 1 est occupee. Le client principal est:
Andree
Voici la commande passee par les clients :
```

PLAT ----Poisson - 60 \$ (20\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :9\$

La table numero 2 est occupee. Le client principal est:
Moussa

Voici la commande passee par les clients :

PLAT ----Poisson - 60 \$ (20\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :9\$

La table numero 3 est occupee. Le client principal est:
Marie

Voici la commande passee par les clients :

PLAT ----Muffin - 5 \$ (2\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :1\$
ET PLAT VEGE Muffin vitamines 9 Proteines 9 Mineraux 1
PLAT ----Pizza - 7 \$ (2\$ pour le restaurant)
PLAT VEGE Pizza vitamines 8 Proteines 6 Mineraux 8
(Apport nutritif 4.8mg)

La table numero 4 est occupee. Le client principal est:
Martin

Voici la commande passee par les clients :

PLAT ----Pates - 30 \$ (9\$ pour le restaurant)

La table numero 5 est vide.

La table numero 6 est occupee. Le client principal est:
John

Voici la commande passee par les clients :

PLAT ----Pates - 30 \$ (9\$ pour le restaurant)

-Voici son menu :

Matin :

PLAT ----Soupe - 10 \$ (5\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :1.5\$
ET PLAT VEGE Soupe vitamines 2 Proteines 3 Mineraux 1
PLAT ----Oeuf - 12 \$ (4.5\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :2.3\$
PLAT ----Pain - 5 \$ (2\$ pour le restaurant)
PLAT VEGE Pain vitamines 6 Proteines 7 Mineraux 2
(Apport nutritif 16.8mg)
PLAT ----Crepes - 6 \$ (2\$ pour le restaurant)
PLAT VEGE Crepes vitamines 9 Proteines 8 Mineraux 1
(Apport nutritif 57.6mg)
PLAT ----Gratin - 23 \$ (8\$ pour le restaurant)
PLAT ----Pancakes - 7 \$ (2\$ pour le restaurant)
PLAT ----Homous - 8 \$ (3\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :0.8\$
ET PLAT VEGE Homous vitamines 1 Proteines 2 Mineraux 3

MENU ENTIEREMENT VEGETARIEN

PLAT VEGE Soupe vitamines 2 Proteines 3 Mineraux 1
PLAT VEGE Pain vitamines 6 Proteines 7 Mineraux 2
PLAT VEGE Crepes vitamines 9 Proteines 8 Mineraux 1
PLAT VEGE Homous vitamines 1 Proteines 2 Mineraux 3

Le plat le moins cher est : PLAT ----Pain - 5 \$ (2\$ pour le restaurant)
PLAT VEGE Pain vitamines 6 Proteines 7 Mineraux 2
(Apport nutritif 16.8mg)

Midi :

PLAT ----Poulet - 20 \$ (6\$ pour le restaurant)
PLAT ----Nachos - 8 \$ (3\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :1.2\$
ET PLAT VEGE Nachos vitamines 5 Proteines 3 Mineraux 6
PLAT ----Boeuf - 29 \$ (12\$ pour le restaurant)
PLAT ----Burrito - 8 \$ (2\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :1\$
PLAT ----Quesadillas - 9 \$ (4\$ pour le restaurant)
PLAT VEGE Quesadillas vitamines 7 Proteines 9 Mineraux 1
(Apport nutritif 50.4mg)
PLAT ----Ratatouille - 8 \$ (5\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :2\$

MENU ENTIEREMENT VEGETARIEN

PLAT VEGE Nachos vitamines 5 Proteines 3 Mineraux 6
PLAT VEGE Quesadillas vitamines 7 Proteines 9 Mineraux 1

Le plat le moins cher est : PLAT ----Nachos - 8 \$ (3\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :1.2\$
ET PLAT VEGE Nachos vitamines 5 Proteines 3 Mineraux 6

Soir :

PLAT ----Pates - 30 \$ (9\$ pour le restaurant)
PLAT ----Poisson - 60 \$ (20\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :9\$
PLAT ----Poulet - 20 \$ (8\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :1.8\$
PLAT ----Muffin - 5 \$ (2\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :1\$
ET PLAT VEGE Muffin vitamines 9 Proteines 9 Mineraux 1
PLAT ----Pizza - 7 \$ (2\$ pour le restaurant)
PLAT VEGE Pizza vitamines 8 Proteines 6 Mineraux 8
(Apport nutritif 4.8mg)

MENU ENTIEREMENT VEGETARIEN

PLAT VEGE Muffin vitamines 9 Proteines 9 Mineraux 1
PLAT VEGE Pizza vitamines 8 Proteines 6 Mineraux 8

Le plat le moins cher est : PLAT ----Muffin - 5 \$ (2\$ pour le restaurant)
Plat Bio comprend une Taxe ecologique de :1\$
ET PLAT VEGE Muffin vitamines 9 Proteines 9 Mineraux 1

```

Le chiffre d'affaire du restaurant PolyFood est 118.3
=====Livraisons=====
Le client Martin n est pas admissible a la livraison
livraison en cours ...
Statut de la table de livraison:(table numero 5):
La table numero 5 est occupee. Le client principal est:
Marie
Voici la commande passee par les clients :
    PLAT ----Pates - 30 $ (9$ pour le restaurant)
    PLAT ----Poulet - 20 $ (8$ pour le restaurant)
Plat Bio    comprend une Taxe ecologique de :1.8$
    PLAT ----Pizza - 7 $ (2$ pour le restaurant)
PLAT VEGE Pizza vitamines 8 Proteines 6 Mineraux 8
(Apport nutritif 4.8mg)
Le client Marie a ete livre
Le chiffre d'affaire du restaurant PolyFood est 148.7
polyFodd2 gagne plus d'argent
la taxe a payer est 1.652

```

Spécifications générales

- Ajouter un destructeur pour chaque classe chaque fois que cela vous semble pertinent.
- Utiliser la liste d'initialisation pour l'implémentation de vos constructeurs.
- Ajouter le mot-clé *const* chaque fois que cela est pertinent.
- Appliquez un affichage « user friendly » (ergonomique et joli) pour le rendu final.
- Documenter votre code source.
- **Bien lire le barème de correction ci-dessous.**

Questions

Répondez aux questions

1. Donner le diagramme de classe
2. Quel est la différence entre une classe abstraite et une interface?

Correction

La correction du TP se fera sur 20 points.

- (3 points) Compilation
- (3 points) Exécution
- (3 points) Comportement

- (3 points) Utilisation du polymorphisme et de `dynamic_cast`
- (2 points) Documentation du code et bonne norme de codage ;
- (2 points) Utilisation de `const` / du mot clef `this`
- (2 points) Gestion de la mémoire (destructeurs, copies, etc.)
- (2 points) Réponse aux questions.