INF1010

Programmation Orientée-Objet

Travail pratique #1
Allocation dynamique, composition et agrégation

Objectifs :	Permettre à l'étudiant de se familiariser avec les notions de base de la programmation orientée objet, l'allocation dynamique de la mémoire, le passage de paramètres, les méthodes constantes et les principes de relation de composition et d'agrégation
Remise du travail :	Mardi 27 Septembre 2016, 8h
Références :	Notes de cours sur Moodle & Chapitre 2-7 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. Les travaux dirigés s'effectuent obligatoirement en équipe de deux personnes faisant partie du même groupe. Veuillez suivre le guide de codage

Informations préalables

La directive de précompilation « #ifndef »

La directive de précompilation « #ifndef » signifie « if not defined » (si non défini). Comme le type de directive le laisse deviner, cette directive est évaluée avant la phase de compilation du code source. Dans les travaux pratiques, vous l'utiliserez dans les fichiers d'en-têtes (.h), pour éviter la double inclusion. Un fichier peut inclure deux fichiers d'entête, par exemple prenons deux fichiers a.h et b.h. Il se peut que a.h soit inclus dans le fichier b.h. On se retrouve alors à inclure deux fois le fichier a.h, ce qui entrainerait une erreur de compilation, car on ne peut définir deux fois la même classe. La directive « #ifndef » nous évite cette double inclusion. Pour utiliser la directive « #ifndef », il faut respecter la syntaxe suivante :

```
#ifndef NOMCLASSE_H

#define NOMCLASSE_H

// Définir la classe NomClasse ici

#endif
```

La directive de précompilation « #include»

La directive de précompilation pour l'inclusion de fichiers « #include »

1. #include <nom_fichier>
2. #include "nom_fichier"

Ce qui différencie ces deux expressions est l'emplacement où le fichier spécifié est recherché. Pour la seconde forme, le précompilateur commence tout d'abord par rechercher dans le même répertoire que le fichier compilé. Par la suite, il procède de la même manière que la première forme, c'est-à-dire dans des répertoires prédéfinis par l'environnement de développement intégré.

En résumé, lorsqu'on inclut un fichier source qui se trouve dans le projet, on utilise la seconde forme. Et lorsque l'on inclut un fichier qui provient d'une bibliothèque externe au projet, on utilise la première forme.

Travail à réaliser

Le travail consiste à implémenter un programme pouvant servir de base pour la réalisation d'un jeu qui sera développé tout au cours de la session.

Bienvenue dans le pays Fabuleux de Polyland, ce pays qui regorge de créatures en tout genre dotées de pouvoirs qui leur sont propres. Dans ce monde, les dresseurs peuvent se déplacer de pays en pays et sont capables de capturer des créatures. Les créatures sont uniques à chaque dresseur et spécifique à chaque pays comme Polyland.

Vous devez réaliser la définition et l'implémentation des classes *Dresseur*, *Pouvoir, ObjetMagique*, *Creature*, Poly*land*.

Pour vous aider, le fichier d'en-tête (sans les commentaires) de la classe Dresseur vous est fourni. Vous devrez implémenter les méthodes du fichier source (.cpp) correspondant ainsi que les classes complètes manquantes.

Note : Lorsque l'on vous demande d'initialiser les attributs aux valeurs par défaut, initialisez :

- Un attribut String a un string vide
- Un entier a la valeur 0
- Un pointeur en le faisant pointer sur nullptr

Pour chaque attribut, vous devrez déterminer vous-même s'il s'agit d'une agrégation ou d'une composition puis faire l'implémentation de la manière appropriée.

Classe *Dresseur*

Créer une classe *Dresseur* qui représente un joueur du monde de PolyLand.

Cette classe contient les attributs suivants :

- Un nom (string)
- Un Tableau de pointeurs de Creature .
- Un nombre de créatures (entier positif)
- Un nombre de créatures maximum, qui servira à connaître la taille du tableau de pointeurs, sa taille initiale sera de 2 (entier)
- Un objet magique (ObjetMagique)

Les méthodes suivantes doivent être implémentées :

- Un constructeur par défaut qui initialise les trois attributs aux valeurs par défaut.

- Un constructeur qui prend un paramètre un nom et qui initialise l'attribut correspondant. Les autres attributs seront initialisés par défaut.
- Un destructeur.
- Les méthodes d'accès et de modification des attributs.
- Une méthode *ajouterCreature()* qui prend en paramètre une créature et qui l'ajoute au tableau de créatures du dresseur. Si la taille des créatures atteint est égale à la taille maximale, le nombre de créatures maximal va être doublé . L'ajout de la créature ne se fait que si le dresseur ne la possède pas encore. Cette méthode doit renvoyer *true* si l'opération est un succès, *false* sinon.
- Une méthode *retirerCreature()* qui prend en paramètre un nom et retire la créature avec le nom associé si le dresseur la possède. Cette méthode doit renvoyer *true* si l'opération est un succès, *false* sinon.
- Une méthode *utiliserObjetMagique()* qui prend en paramètre un pointeur sur une créature. Cette méthode va appliquer le bonus de l'attribut objetMagique du dresseur à une créature. L'application de l'objet magique sur la créature va augmenter l'attribut point de vie et l'attribut energie de la créature selon la valeur du bonus de l'objetMagique. **Ces deux attributs ne doivent jamais dépasser le nombre les attributs maximaux correspondants.**
- Une méthode *affichage()* qui affiche le nom du dresseur ainsi que le nombre de créatures qu'il possède.

Classe *Creature*

Créer une classe *Creature* qui représente une créature dans le monde de Polyland

Cette classe possède les attributs suivants :

- Un nom (string)
- Une valeur attaque (entier)
- Une valeur defense (entier)
- Un attribut pointDeVie (entier) qui correspond à la valeur courante de point de vie de la créature
- Un attribut pointDeVieTotal (entier) qui correspond à la valeur maximale de point de vie que peut avoir la créature
- Un attribut energie (entier) qui correspond à la valeur courante d'énergie de la créature
- Un attribut energieTotal (entier) qui correspond à la valeur maximale de l'énergie de la créature
- Un attribut experience (entier) qui correspond à l'expérience actuel de la créature
- Un attribut experienceNecessaire (entier) qui correspond à l'expérience nécessaire pour atteindre le prochain niveau
- Un niveau (entier)
- Un pouvoir (Pouvoir)

La classe possède les méthodes suivantes:

- Un constructeur par défaut qui initialise les attributs aux valeurs par défaut.
- Un constructeur par paramètres qui initialise les attributs nom, attaque, defense, pointDeVie, énergie, et pouvoir selon les paramètres correspondants. Les autres paramètres sont initialisés à des valeurs par défaut. Les paramètres pointDeVieTotal et energieTotal sont initialisés aux valeurs de pointDeVie et d'energie, le niveau est intialisé à 1 et l'expérienceNécessaire à 100.
- Un destructeur.
- Les méthodes d'accès et de modification des attributs.
- Une méthode *attaquer()* qui prend en paramètre une créature qui va subir l'attaque. **Vous devez compléter la méthode fournie.**
- Une méthode *experienceGagnee()* qui va calculer et donner l'expérience gagnée lors d'un combat avec une autre créature. **Cette méthode vous est fournie.**
- Une méthode *information()* qui affiche les informations sur la créature, tel que présenté dans l'exemple à la fin du document.

Classe *Pouvoir*

La classe *Pouvoir* sert à représenter les pouvoirs d'une créature du monde de Polyland. Un pouvoir demande une certaine quantité d'énergie pour être exécuté et inflige un certain nombre de dégâts à une autre créature cible.

Cette classe possède les attributs suivants :

- Un nom (string)
- Un nombre de dégâts (entier)
- Une valeur d'énergie nécessaire (entier)

Les méthodes suivantes doivent être implémentées :

- Un constructeur par paramètres qui initialise les attributs avec les paramètres correspondants.
- Un destructeur.
- Les méthodes d'accès et de modification des variables membres.
- Une méthode d'affichage des informations, nommée *description()*, tel que présenté dans l'exemple à la fin du document.

Classe ObjetMagique

La classe *ObjetMagique* sert à représenter les objets magiques utilisables sur une créature du monde de Polyland.

Cette classe va contenir les attributs suivants :

- Un nom (string)

- Un bonus (Entier)

Les méthodes suivantes doivent être implémentées :

- Un constructeur par paramètres qui initialise les attributs avec les paramètres correspondants.
- Un destructeur.
- Les méthodes d'accès et de modification des variables membres.
- Une méthode d'affichage des informations, nommée *affichage(),* tel que présenté dans l'exemple à la fin du document.

Classe *Polyland*

La classe *Polyland* est celle qui fait le lien entre toutes les classes précédentes

Elle contient les attributs suivants :

- Tableau de pointeurs de Dresseurs: maximum de 100 dresseurs
- Nombre de dresseurs (entier)
- Tableau de pointeurs de Creatures: maximum de 1000 créatures
- Nombre de créatures (entier)

Note : Tous les tableaux de pointeurs doivent être alloués dynamiquement.

Elle doit contenir les méthodes suivantes :

- Un constructeur par défaut qui initialise les tableaux avec les dimensions indiquées et initialise les pointeurs à nullptr.
- Un destructeur.
- La méthode *ajouterDresseur()* qui permet d'ajouter dresseur reçu en paramètre , deux dresseurs ne peuvent pas avoir le même nom.
- La méthode *retirerDresseur()* qui permet de retirer le dresseur en utilisant le nom reçu en paramètre.
- La méthode *ajouterCreature()* qui permet d'ajouter la créature reçue en paramètre, deux créatures ne peuvent pas avoir le même nom. *Pensez à faire une copie de la créature*
- La méthode *retirerCreature()* qui permet de retirer la créature en utilisant le nom reçu en paramètre.
- La méthode *choisirDresseurAleatoire()* qui retourne un dresseur choisi aléatoirement dans le tableau de dresseurs. *Aide : Utilisez la fonction srand() et rand(),* http://www.cplusplus.com/reference/cstdlib/srand/
- La méthode *choisirCreatureAleatoire()* qui retourne une créature choisie aléatoirement dans le tableau de créatures.
- La méthode *attraperCreature()* qui prend en paramètre un dresseur et une créature, elle renvoie true si la créature a bien été capturée par le dresseur, false sinon. *Chaque dresseur possède une version unique de ses créatures.*
- La méthode *relacherCreature*() qui prend en paramètre un dresseur et le nom d'une créature, elle renvoie true si la créature a bien été relâchée par le dresseur, false sinon.

- La méthode « *infoDresseur*() » prend en paramètre un nom de dresseur et affiche les informations qui le concernent, tel que présenté dans à la fin du documents
  Aide : Pensez à utiliser les différentes méthodes pour afficher les informations des différentes classes.

Main.cpp

Le programme principal contient des directives à suivre pour instancier différents objets et essayer les différentes méthodes implémentées. **Faites attention au nom que vous donnez à vos méthodes, cela peut engendrer des affichages erronés.**

Le résultat final devrait être similaire à ce qui suit, il se peut que vous ayez une légère différence à cause des choix aléatoires :

```
CREATION DES DRESSEURS

CREATION ET AFFICHAGE DES CREATURES
Pokachu a 10 en attaque et 2 en defense,
Il a 50/50 PV et 25/25 Energie
Il est au niveau 1 avec 0d'XP
Il lui manque 100 jusqu'au prochain niveau
Son pouvoir  est : Eclair necessite 5 mana et inflige 10 degats


Ajout des creatures et dresseurs de Polyland
Salimouche a bien été ajouté !
Carapouce a bien été ajouté !
Balbazar a bien été ajouté !
Pokachu a bien été ajouté !
Regis a bien été ajouté !
Pierre a bien été ajouté !
Sasha a bien été ajouté !

BIENVENUE A POLYLAND

Choix de la premiere creature
Vous etes chanceux voici un Pokachu rien que pour vous

TESTS DE COMBAT

Un Salimouche surgit
Vous avez rencontré un Salimouche sauvage qui vous attaque...
Salimouche lance Boule de feu qui inflige 20 degat a Pokachu
Pokachu a encore 30 PV
Pokachu lance Eclair qui inflige 20 degat a Salimouche
Salimouche a encore 25 PV
Attaque Eclair a échouée
Pokachu lance Eclair qui inflige 20 degat a Salimouche
Salimouche a encore 5 PV
Attaque Eclair a échouée
Pokachu lance Eclair qui inflige 20 degat a Salimouche
Pokachu a gagné 43 XP
Salimouche a encore 0 PV
Vous avez battu un Salimouche, vous pouvez maintenant le capturer
Felicitation vous avez attrapé un Salimouche !

Vous trouvez une potion magique, vous décidez de l'utilisez sur Pokachu

Un Pokachu se jette sur votre Pokachu
Un duel entre Pokachu et Pokachu est engagé
Pokachu lance Eclair qui inflige 30 degat a Pokachu
Pokachu a encore 20 PV
Pokachu lance Eclair qui inflige 30 degat a Pokachu
```

```
Pokachu a encore 15 PV
Attaque Eclair a échouée
Pokachu lance Eclair qui inflige 30 degat a Pokachu
Pokachu a gagné 25 XP
Pokachu a encore 0 PV
Salimouche lance Boule de feu qui inflige 20 degat a Pokachu
Pokachu a encore 0 PV
Votre Pokachu a été battu mais heuresement votre Salimouche finit par vaincre Pokachu
Vous essayez d'attraper Pokachu
Pokachu s'est échappé

AFFICHAGE DE VOS INFORMATIONS
Vous a bien été ajouté !
Informations sur le dresseur:
Vous possede 2 creature(s)
- 1 - Pokachu a 10 en attaque et 2 en defense,
Il a 0/50 PV et 20/25 Energie
Il est au niveau 1 avec 43d'XP
Il lui manque 57 jusqu'au prochain niveau
Son pouvoir  est : Eclair necessite 5 mana et inflige 10 degats

- 2 - Salimouche a 12 en attaque et 3 en defense,
Il a 0/45 PV et 10/20 Energie
Il est au niveau 1 avec 0d'XP
Il lui manque 100 jusqu'au prochain niveau
Son pouvoir  est : Boule de feu necessite 5 mana et inflige 5 degats

Pokachu et salimouche n'arrete pas de se chamailler, vous decidez d'abandonner Salimouche
Informations sur le dresseur:
Sasha possede 1 creature(s)
- 1 - Salimouche a 12 en attaque et 3 en defense,
Il a 45/45 PV et 20/20 Energie
Il est au niveau 1 avec 0d'XP
Il lui manque 100 jusqu'au prochain niveau
Son pouvoir  est : Boule de feu necessite 5 mana et inflige 5 degats

Vous avez decidé de relacher Salimouche !

UN DRESSEUR SOUHAITE VOUS DEFIEZ
Pierre possede 1 creature(s)
Pierre surgit avec son Balbazar
Un duel entre Pokachu et Balbazar est engagé
Pokachu lance Eclair qui inflige 30 degat a Balbazar
Balbazar a encore 20 PV
Balbazar lance Lance feuille qui inflige 15 degat a Pokachu
Pokachu a encore 35 PV
Pokachu lance Eclair qui inflige 30 degat a Balbazar
Pokachu a gagné 35 XP
Balbazar a encore 0 PV
```

```
Suite à sa défaite Pierre s'enfuit de Polyland
Pierre s'est bien enfuit de Polyland...

INFO DRESSEUR APRES TOUTES CES PERIPETIES
Informations sur le dresseur:
Vous possede 1 creature(s)
- 1 - Pokachu a 10 en attaque et 2 en defense,
Il a 35/50 PV et 15/25 Energie
Il est au niveau 1 avec 78d'XP
Il lui manque 22 jusqu'au prochain niveau
Son pouvoir  est : Eclair necessite 5 mana et inflige 10 degats

Dresseur introuvable!
Appuyez sur une touche pour continuer...
```

Question

1. Quel est le lien (agrégation/composition) entre les dresseurs et Polyland ? Justifiez votre choix de lien.
2. Quel est le lien (agrégation/composition) entre les créatures et Polyland ? Justifiez votre choix de lien.
3. Quel est le lien (agrégation/composition) entre les dresseurs et les créatures ? Justifiez votre choix de lien.
4. Quel est le lien (agrégation/composition) entre les créatures et les pouvoirs ? Justifiez votre choix de lien.

Correction

La correction du TP1 se fera sur 20 points.

Voici les détails de la correction :

- (3 points) Compilation du programme ;
- (3 points) Exécution du programme ;
- (4 points) Comportement exact des méthodes du programme ;
- (2 points) Documentation du code et bonne norme de codage ;
- (2 points) Utilisation correcte du mot-clé *const* et dans les endroits appropriés ;
- (2 points) Utilisation adéquate des directives de précompilation ;
- (2 points) Allocation et désallocation appropriées de la mémoire ;
- (2 points) Réponse aux questions ;