

TD1- Conception et implémentation de POO

Objectifs du TD

- Pratiquer l'héritage simple afin de bien comprendre cette notion.
- Savoir modéliser un programme orienté objets grâce à UML
- Bonnes pratiques de développement :
 - ✓ Commentaires
 - ✓ Nomination des attributs
 - ✓ Utiliser un jeu de test pour vérifier les fonctionnalités d'un objet

Exercice 1 :

Pour gérer les appareils électroniques dans un centre de formation en électronique, le directeur du centre propose de créer un programme orienté objets en se basant sur les informations suivantes :

Chaque appareil est caractérisé par un identifiant et un nom. Parmi les appareils, on trouve les voltmètres qui sont des appareils ayant une tension maximale. On trouve aussi les ampèremètres qui ont une intensité maximale. Enfin, il y a les oscilloscopes qui ont une marque et une taille qui peut être petit, moyen ou grand.

On souhaite que le programme puisse ajouter un appareil, supprimer un appareil, et afficher le descriptif des appareils détenus en stock.

T.A.F

- 1- Tracer le diagramme de classe modélisant le programme.
- 2- Implémenter les classes en C#.
- 3- Tester le programme.

Exercice 2

On se propose de créer un programme orienté objets pour gérer une bibliothèque selon les informations suivantes :

- La bibliothèque constituée d'étagères, sur lesquelles sont déposés des documents. Tous les documents ont un numéro d'enregistrement (un entier) et un titre (une chaîne de caractères). Les Livres ont, en plus, un auteur (une chaîne) et un nombre de pages (un entier). Les romans (qui sont des livres)

ont éventuellement un prix littéraire (chaîne), tandis que les manuels (aussi des livres) ont un niveau scolaire (entier). Les revues sont des documents avec un mois et une année (entiers). Les dictionnaires sont des documents avec une langue (chaîne).

- On pourra ajouter une étagère ou un document à la bibliothèque. On considèrera qu'une étagère a un nombre maximum de documents gérables. L'ajout d'un document ne pourra se faire que s'il reste une place disponible pour celui-ci.
- On pourra rechercher un document, un livre, une revue, etc. dans la bibliothèque.

T.A.F

- 1- Tracer le diagramme de classe modélisant le programme.
- 2- Implémenter les classes en C#.
- 3- Tester le programme.

Exercice 3 :

L'objectif de cet exercice est de créer un programme orienté objet d'un jeu de guerre en mode Console. Ce jeu repose sur deux principaux éléments : Personnage et Arme.

Toute arme possède un nom et un niveau d'attaque (un entier ≤ 100). On doit pouvoir initialiser ses 2 paramètres à la création de l'arme. On doit pouvoir connaître le niveau d'attaque. On doit pouvoir visualiser (dans la console) le nom et le niveau d'attaque. On doit pouvoir avoir accès au nom de l'arme.

Parmi les armes, on trouve essentiellement l'Épée et le Bâton.

L'épée possède un indice (< 100) de finesse alors que le bâton un âge (< 100). Ces 2 paramètres doivent être initialisés dès la création d'un bâton ou d'une épée.

Quant au personnage, tout personnage possède un nom et un niveau de vie. On doit pouvoir initialiser ses 2 paramètres à la création du personnage. On doit pouvoir connaître le niveau de vie. On doit pouvoir visualiser (dans la console) le nom et le niveau de vie. On doit pouvoir avoir accès au nom du personnage.

Parmi les personnages du jeu, on trouve le Guerrier et le Magicien.

Un guerrier peut être à cheval. Un magicien peut être novice ou confirmé. Ces 2 paramètres doivent être initialisés dès la création d'un personnage.

Les règles du jeu sont les suivantes :

Un personnage peut posséder jusqu'à 5 armes. On peut donc ajouter une arme à un personnage (mais pas en retirer). On doit pouvoir affecter une arme à un personnage qui va l'utiliser.

Un personnage peut se fatiguer (suite à un combat, un déplacement, etc.), ce qui lui enlèvera un certain nombre de points de vie. La fatigue d'un Magicien par exemple lui fait toujours perdre 10 points. La fatigue d'un Guerrier lui fait toujours perdre 20 points. Un personnage est vivant si son niveau de vie est positif. Pour tester le jeu, on va créer une liste de 3 personnages : 2 Magiciens et 1 Guerrier. Ensuite, ajouter 3 armes au guerrier et faites lui choisir la seconde. Enfin, fatiguer le guerrier et les Magiciens.

T.A.F

1- Tracer le diagramme de classe modélisant le programme.

2- Implémenter les classes en C#.

3- Tester le programme.