TD5- Conception et implémentation de POO

Objectifs du TD

- Pratiquer l'héritage avec des classes abstraites 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 :

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'Epée 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 digramme de classe modélisant le programme.

2- Implémenter les classes en C#.

3- Tester le programme.

Exercice 2 :

Un directeur de magasin commercial souhaite automatiser la gestion de ses ventes à travers la création d'un POO. Le magasin vend 3 types de produit : alimentaire, électroménager et cosmétique. Un produit est caractérisé par une référence, un libellé, une quantité en stock, un prix d'achat et un prix de vente hors taxe. Les produits alimentaires ont comme caractéristique supplémentaire une date limite de consommation. Pour les produits électroménagers, il faut spécifier la taille (petit, moyen ou grand) ainsi que la marque. Enfin, un produit cosmétique peut être liquide ou solide.

Le magasin s'approvisionne de 3 fournisseurs (fournisseur de produit alimentaire, fournisseur de produit électroménager et fournisseur de produit cosmétique). Dans une opération d'approvisionnement, le fournisseur précise le prix de vente TTC du produit au magasin ainsi que la possibilité de fournir la quantité demandée. Dans le cas où le fournisseur est capable de fournir la quantité demandée, on augmente la quantité en stock du produit commandé. Dans le cas inverse, on attend jusqu'à ce que le fournisseur fournisse la quantité de produit demandée.

La gestion du magasin se fait selon les règles suivantes :

On doit pouvoir afficher le nombre total de produits dans le magasin.

On doit pouvoir calculer les prix de vente TTC des produits sachant que les produits alimentaires, les produits électroménagers et les produits cosmétiques sont assujettis à une TVA de 18%, 20% et 22% et que les marges respectives sont 15%, 18% et 23%.

On doit pouvoir vendre des produits si la quantité demandée est inférieure à la quantité en stock.

On doit pouvoir s'approvisionner auprès des fournisseurs.

On doit pouvoir calculer le bénéfice net du magasin.

On doit pouvoir comparer deux produits.

T.A.F

1- Tracer le digramme de classe modélisant le programme.

2- Implémenter les classes en C#.

3- Tester le programme.