Examen de Rattrapage POO - 2017 (01H30)

Exercice 1.

Soit l'interface:

```
interface Forme{

    Forme combine(Forme forme); //combine deux Forme

    Forme cassé();

}
```

Q1. Ecrire la classe Lego suivante:

Un Lego a une couleur (int) et une forme (Forme). On construit un objet Lego en fournissant la couleur et la forme. On peut construire aussi un Lego en fournissant seulement une forme, dans ce cas la couleur sera égale à 1. Ecrire ce deuxième constructeur en une seule instruction.

La classe dispose d'une méthode qui peut joindre le Lego sur lequel elle est invoquée avec un autre objet Lego. Le résultat sera un nouvel objet Lego de couleur 1 et de forme combinée des deux formes.

La classe dispose d'une méthode qui divise le Lego, elle renvoi une collection de deux (02) Lego: le premier Lego de la collection est identique au Lego sur lequel est invoquée la méthode; le deuxième Lego a la même couleur et la forme cassé du Lego sur lequel est invoquée la méthode.

Exercice 2.

Un employé à un numéro (int), un salaire de base (int) et un nombre d'années de service (int). Ces trois attributs sont **private**. Un constructeur pour initialiser tous les attributs et une méthode de calcul du salaire avec la formule: salaire de base * années de service.

En plus de ces employé "standard", il y'a des employé de luxe. Ils sont exactement comme les employé standards mais chaque employé de luxe a un bonus et sa paye est calculée avec la formule: salaire de base * années de service * bonus. De plus, on a une méthode qui change le bonus d'un employé de luxe.

Q1. Ecrire les deux classes en utilisant l'héritage.

Q2. Ecrire la classe Personnels qui dispose d'une collection d'Employé. Et de deux méthodes:

calculAllSalaire(): invoque le calcul de salaire de tous les employé.

nouveauBonus(int nb): pour tous les employé de luxe, le bonus est mis à jour.

```
Aide: ArrayList<Desi> elements; // déclaration du référent
elements = new ArrayList<Desi>(); // création d'objet ArrayList typé

elements.add(...); //ajout element
int nbelements.size();// récupère le nombre d'elements dans la liste

                ... valeur = elements.get(i); // récupère l'élément d'indice i
for(Desi t:elements) {.. t.dst()...}
```

UNIVERSITE DU 8 MAI 1945 Faculté des Mathématiques et de l'Informatique Et des Sciences de la Matière DEPARTEMENT D'INFORMATIQUE

D.Bouroualeh 2017 – 2ème Li

Examen de Rattrapage POO - 2017 (01H30)

Exercice 1.

Soit l'interface:

```
interface Forme{

    Forme combine(Forme forme); //combine deux Forme

    Forme cassé();

}
```

Q1. Ecrire la classe Lego suivante:

Un Lego a une couleur (int) et une forme (Forme). On construit un objet Lego en fournissant la couleur et la forme. On peut construire aussi un Lego en fournissant seulement une forme, dans ce cas la couleur sera égale à 1. Ecrire ce deuxième constructeur en une seule instruction.

La classe dispose d'une méthode qui peut joindre le Lego sur lequel elle est invoquée avec un autre objet Lego. Le résultat sera un nouvel objet Lego de couleur 1 et de forme combinée des deux formes.

La classe dispose d'une méthode qui divise le Lego, elle renvoi une collection de deux (02) Lego: le premier Lego de la collection est identique au Lego sur lequel est invoquée la méthode; le deuxième Lego a la même couleur et la forme cassé du Lego sur lequel est invoquée la méthode.

Exercice 2.

Un employé a un numéro (int), un salaire de base (int) et un nombre d'années de service (int). Ces trois attributs sont **private**. Un constructeur pour initialiser tous les attributs et une méthode de calcul du salaire avec la formule: salaire de base * années de service.

En plus de ces employé "standard", il y'a des employé de luxe. Ils sont exactement comme les employé standards mais chaque employé de luxe a un bonus et sa paye est calculée avec la formule: salaire de base * années de service * bonus. De plus, on a une méthode qui change le bonus d'un employé de luxe.

Q1. Ecrire les deux classes en utilisant l'héritage.

Q2. Ecrire la classe Personnels qui dispose d'une collection d'Employé. Et de deux méthodes:

calculAllSalaire(): invoque le calcul de salaire de tous les employé.

nouveauBonus(int nb): pour tous les employé de luxe, le bonus est mis à jour.

Aide:

```
ArrayList<Desi> elements; // déclaration du référent
elements = new ArrayList<Desi>(); // création d'objet ArrayList typé

elements.add(.....); //ajout element
int nbelements.size();// récupère le nombre d'elements dans la liste

                // value = elements.get(3); // récupère l'élément d'indice 3
for(Forme t:Plateau) {.. t.dat();....}
```