جامعة مولاي إسماعيل UNIVERSITÉ MOULAY ISMAÏL المدرسة العليا للأساتذة ÉCOLE NORMALE SUPÉRIEURE	**TP3**: PROGRAMMATION ORIENTÉE OBJET **JAVA** ➢ **Manipulation des classes et des objets** ➢ **Utilisation de méthodes statiques**	**LEI (S5)** **PAR MR** : M.MOUKHAFI

Exercice 1 : Articles à vendre

Une société vend des articles de papeterie. Vous vous limiterez aux articles suivants :

- stylos décrits par une référence, un nom ou descriptif (par exemple "stylo noir B2"),une marque, un prix unitaire et une couleur,

- ramettes de papier décrites par une référence, un nom, une marque, un prix unitaire et le grammage du papier (par exemple, 80 g/m²).

De plus cette société peut vendre ces articles par lots. Vous supposerez que les lots sont composés d'un certain nombre d'un même type d'articles, par exemple un lot de 10 stylos noirs B2 de la marque WaterTruc. Un lot a une référence. Il a une marque qui est celle de l'article dont il est composé (WaterTruc pour l'exemple précédent). Le nom du lot est déterminé par le nombre d'articles dont il est composé ; par exemple "Lot de 10 stylo noir B2" (vous êtes dispensé de mettre les noms au pluriel !).

Le prix du lot est déterminé par le prix unitaire de l'article multiplié par le nombre d'articles, auquel est enlevé un certain pourcentage qui dépend du lot. Par exemple, si un lot de 10 stylos à 100 DH a un pourcentage de réduction de 20 %, le prix du lot sera de 10 x 100 x (100 - 20) / 100 = 800 DH. Le prix d'un lot sera calculé au moment où on demandera le prix du lot ; il n'est pas fixé une fois pour toute et il change si le prix de l'article qui compose le lot est modifié.

-Ecrivez des classes dans un paquetage "ens.mit.article.toto" pour représenter les différents articles (où vous remplacerez toto par votre nom).

On ne s'intéresse qu'aux différents types d'articles. Ainsi, on aura une seule instance qui représentera le type "stylo noir B2" et pas 250 instances

si la société possède 250 "stylo noir B2". Si vous voulez démarrer une gestion de stock, vous pouvez ajouter le nombre d'articles en stock pour chaque type.

Vos classes devront **permettre** de faire du polymorphisme avec la méthode qui donnera le prix d'un article (application dans l'exercice suivant sur les factures).

Exercice 2 : Du polymorphisme dans une facture

La société établit des factures numérotées (en partant de 1) et datées, comprenant le nom du client, le prix total de la commande et, pour chaque article commandé, sa référence, son prix unitaire, le nombre d'articles commandés et le prix total pour cet article.

Pour simplifier, un client ne sera qu'un simple nom dans la facture (pas de classe Client).

Ajoutez les classes pour représenter ces factures, dans un paquetage " ens.mit.toto.facture" (où vous remplacerez toto par votre nom).

Vous écrirez une classe test qui affiche 1 facture pour
10 stylos (choisissez ses caractéristiques)

15 ramettes de papier

2 lots de 5 stylos (différents des 10 stylos)

et une autre facture pour

3 stylos

2 ramettes

pour 2 clients différents.

Le programme créera tous les objets nécessaires à l'affichage des 2 factures.

Une facture s'affichera sur le modèle suivant (on ne demande pas d'affiner davantage le format d'affichage mais si vous pouvez faire mieux, n'hésitez pas !) :

```
------------------------------------------------------------------
Facture 125  Client: Toto  Date: 10 novembre 1999
Quant. Ref.   Nom            Marque     PU      PT
10       s1     Stylo Jade       Watertruc      500.0  5000.0
15       r1     Ramette haute qualité  Clairefont    95.0    1425.0
2        l1     Lot de 5 Stylo or    Marker 4500.0  9000.0
Prix total facture : 15425.0 DH
------------------------------------------------------------------
```

Exercice 5 : L'héritage chez les animaux

Des animaux, des mammifères et des poissons. Des chiens et des lions...
Modélisez avec des classes. Ajoutez à chaque classe fille au moins une variable et une méthode nouvelles. Arrangez-vous pour que les classes filles puissent manipuler directement les nouvelles variables.

Voici ci-dessous une classe de test et l'affichage que vos classes devront fournir. La méthode getType de vos classes devra fournir une description de l'instance, en incluant les descriptions de toutes les classes mères. Pour cela, il vous est imposé d'utiliser le mot-clé super.

La classe de test :

```
public class TestAnimal {
public static void main(String[] args) {
Animal[] animaux = new Animal[5];
animaux[0] = new Animal("Truc");
animaux[1] = new Animal();
animaux[2] = new Chien("Médor");
animaux[3] = new Lion();
animaux[4] = new Lion ("Robert");
for (int i = 0; i < animaux.length; i++) {
System.out.println(animaux[i].getType());
}
}
}
```

L'affichage :

Je suis un animal de nom Truc.

Je suis un animal.

Je suis un animal de nom Médor. Je suis un mammifère.

Je suis un chien.

Je suis un animal. Je suis un mammifère.

Je suis un lion.

Je suis un animal de nom Robert.

Je suis un mammifère. Je suis un lion.