INSTITUT SUPERIEUR DES ETUDES TECHNOLOGIQUES DE NABEUL

TRAVAUX PRATIQUES N° 8 : LES CLASSES ABSTRAITES ET LES INTERFACES

Classe : DSI 2 **Enseignante : Sana REFAI**

1. Objectifs

Le but de ce TP est d'assimiler l'intérêt d'une classe abstraite et d'apprendre à les exploiter ainsi que de maîtriser l'utilisation des interfaces pour l'héritage multiple.

Note importante :
Chaque classe doit être déclarée dans un fichier indépendant portant son nom.

2. Les Exercices

2.1 Exercice 1

IL s'agit de créer un programme Java pour gérer une collection d'ustensiles de cuisine anciens[4], plus précisément des assiettes (rondes ou carrées) et des cuillères. Vous trouverez ci-dessous une version incomplète de son programme Collection. Les 5 objets créés sont stockés dans le tableau us qui est de type Ustensile[].

Il vous est demandé de compléter le programme selon les indications dans les quatre questions ci-après. Pensez à profiter au maximum des possibilités offertes par Java pour éviter la duplication inutile d'instructions dans les classes et les méthodes.

```
class Collection {


public static void main(String[] args) {

Ustensile[] us = new Ustensile[5];

us[0] = new AssietteRonde(1926, 8.4);

us[1] = new Cuillere(1881, 7.3);

us[2] = new AssietteCarree(1935, 5.6);

us[3] = new Cuillere(1917, 8.8);

us[4] = new AssietteRonde(1837, 5.4);


afficherCuilleres(us);

afficherSurfaceAssiettes(us);
```

```
afficherValeurTotale(us);

}


static void afficherCuilleres(Ustensile[] us) { int nbCuilleres = 0;

for (int i = 0; i < us.length; i++) {

// A compléter

}

System.out.println("Il y a " + nbCuilleres + " cuillères.");

}
```

Hiérarchie de classes

Il convient de modéliser les objets de la collection avec une hiérarchie de 5 classes comme indiqué dans la liste ci-dessous. Ecrivez le code de cette hiérarchie de classes, y compris les variables d'instance et les constructeurs. Vous pouvez ajouter les nouvelles classes au fichier Collection.java, si vous voulez, ou bien utiliser des fichiers séparés. Evitez de dupliquer inutilement des instructions. La méthode main du programme Collection ci-dessus vous montre la façon d'appeler les constructeurs des classes qui sont instanciables.

1. Chaque Ustensile a une annee de fabrication qui est une valeur int. Un Ustensile est soit une Assiette, soit une Cuillere. Il n'est pas possible d'instancier la classe Ustensile.

2. Une Assiette est soit une AssietteRonde, soit une AssietteCarree. Il n'est pas possible d'instancier la classe Assiette.

3. Chaque AssietteRonde à un rayon qui est une valeur double.

4. Chaque AssietteCarree à un cote qui est une valeur double.

5. Chaque Cuillere à une longueur qui est une valeur double.

6.

Comptage

Complétez le code de la méthode afficherCuilleres pour qu'elle calcule et affiche le nombre d'objets de type Cuillere qui sont stockés dans le tableau us. Par exemple, le code de la méthode main ci-dessus donne lieu à l'affichage suivant:

Il y a 2 cuillères.

Surface totale

Complétez le code de la méthode afficherSurfaceAssiettes pour qu'elle calcule et affiche la somme des surfaces de toutes les assiettes stockées dans le tableau us, c'est-à-dire les assiettes rondes et les assiettes carrées. Il sera nécessaire de compléter la hiérarchie des classes avec des méthodes de calcul de surface. Evitez de dupliquer inutilement des instructions La surface d'une assiette se calcule comme suit:

1. AssietteRonde : 3.14 * rayon * rayon

2. AssietteCarree : cote * cote

Par exemple, le code de la méthode main ci-dessus donne lieu à l'affichage suivant:

Surface totale des assiettes : 344.48080000000004

Valeur totale des ustensiles

Complétez le code de la méthode afficherValeurTotale pour qu'elle calcule et affiche la somme des valeurs de tous les ustensiles stockés dans le tableau us. Il sera nécessaire de compléter la hiérarchie des classes avec des méthodes de calcul de valeur. Evitez de dupliquer inutilement des instructions.

La valeur d'un ustensile se calcule comme suit :

1. Cuillere et AssietteRonde : Si l'ustensile a moins de 50 ans, il vaut 0 dinars. Sinon, il vaut 1 franc pour chaque année qu'il a de plus de 50 ans, c'est-à-dire (2019 - annee - 50). Par exemple, une AssietteRonde ou une Cuillere fabriquée en 1943 vaut 16 dinars.

2. AssietteCarree : 5 fois la valeur qu'elle aurait eue en étant ronde (car les assiettes carrées sont plus rares). Par exemple, une AssietteCarree fabriquée en 1943 vaut 5 * 16 dinars.

Par exemple, le code de la méthode main ci-dessus donne lieu à l'affichage suivant :

Valeur totale de la collection :

2.2 Exercice 2

Une agence de location de voitures [5]offre à ses clients la possibilité de choisir la voiture louée en fonction de différents critères. Les voitures sont définies par une marque, un nom de modèle, une année de production et un prix de location à la journée. Pour simplifier les deux premiers attributs seront des objets de la classe String et les deux derniers seront des int.

Q1. Donner le code de la classe Voiture pour laquelle on souhaite disposer la méthodes toString qui retourne une chaine de caractères reprenant la marque, le nom du modèle et le prix de location. Il est possible de sélectionner parmi les voitures à louer toutes les voitures satisfaisant un critère donné. On définit l'interface Critère ainsi :

public interface Critere {

/public boolean estSatisfaitPar(Voiture v);}

// La methode estSatisfaitPar return true si et seulement si la voiture est conforme au critère

Q2. Donnez le code d'une classe CritereMarque qui est un critère satisfait par toutes les voitures d'une marque donnée. La marque est précisée la construction du critère (dans le constructeur).

Q3. Donnez le code d'une classe CriterePrix qui est un critère satisfait par toutes les voitures dont le prix est inférieur à un prix fixé à la construction du critère (dans le constructeur).

Q4. On suppose une classe Agence définie (au minimum) ainsi On suppose une classe Agence définie (au minimum) ainsi :

Classe agence {

Voiture [] Voitures ;

Int nbrV ;

Agence (...)

afficheSelection(c : Critere)...

Ajout(Voiture) }

Donnez le code de la méthode afficheSelection(c : Critere) qui affiche parmi toutes les voitures de l'agence (contenues dans l'attribut voitures) celles qui satisfont le critère donné.

Q5. Dans la classe test : Créez une agence et remplir l'agence avec des voitures

- Donnez la ou les lignes de code permettant d'afficher toutes les voitures de cette agence dont le prix est inférieur à 100

- Donnez la ou les lignes de code permettant d'afficher toutes les voitures de cette agence dont la marque est " Clio"

2.3 Exercice 3

Le directeur d'une entreprise de produits chimiques souhaite gérer les salaires et primes de ses employés au moyen d'un programme Java.

Un employé est caractérisé par son nom, son prénom, son âge et sa date d'entrée en service dans l'entreprise.

Dans un fichier Employe.java, codez une classe abstraite Employe dotée des attributs nécessaires, d'une méthode abstraite calculerSalaire (ce calcul dépendra en effet du type de l'employé) et d'une méthode getNom retournant une chaine de caractère obtenue en concaténant la chaine de caractères "L'employé " avec le prénom et le nom.

Dotez également votre classe d'un constructeur prenant en paramètre l'ensemble des attributs nécessaires.

Partie 1 Calcul du salaire

Le calcul du salaire mensuel dépend du type de l'employé. On distingue les types d'employés suivants :

- Ceux affectés à la Vente. Leur salaire mensuel est le 20 % du chiffre d'affaire qu'ils réalisent mensuellement, plus 400 dinars.

- Ceux affectés à la Représentation. Leur salaire mensuel est également le 20 % du chiffre d'affaire qu'ils réalisent mensuellement, plus 800 dinars.

- Ceux affectés à la Production. Leur salaire vaut le nombre d'unités produites mensuellement multipliées par 5.

- Ceux affectés à la Manutention. Leur salaire vaut leur nombre d'heures de travail mensuel multipliées par 65 dinars.

L’ hiérarchie de classes pour les employés en respectant les conditions suivantes :

- La super-classe de la hiérarchie doit être la classe Employe.

- Les nouvelles classes doivent contenir les attributs qui leur sont spécifiques ainsi que le codage approprié des méthodes calculerSalaire et getNom, en changeant le mot "employé" par la catégorie correspondante.

- Chaque sous classe est dotée de constructeur prenant en argument l'ensemble des attributs nécessaires.

- N'hésitez pas à introduire des classes intermédiaires pour éviter au maximum les redondances d'attributs et de méthodes dans les sous-classes.

Partie 2 Employés à risques

Certains employés des secteurs production et manutention sont appelés à fabriquer et

manipuler des produits dangereux. Après plusieurs négociations syndicales, ces derniers

parviennent à obtenir une prime de risque mensuelle.

Complétez votre programme en introduisant deux nouvelles sous-classes d'employés. Ces sous-classes désigneront les employés des secteurs production et manutention travaillant avec des produits dangereux.

Ajouter également à votre programme une interface pour les employés à risque permettant de leur associer une prime mensuelle fixe de 200.

Partie 3 Collection d'employés

Satisfait de la hiérarchie proposée, notre directeur souhaite maintenant l'exploiter pour afficher le salaire de tous ses employés ainsi que le salaire moyen.

Ajoutez une classe Personnel contenant une "collection" d'employés.

Définissez ensuite les méthodes suivantes à la classe Personnel :

- void ajouterEmploye(Employe) qui ajoute un employé à la collection.

- void calculerSalaires() qui affiche le salaire de chacun des employés de la collection.

- double salaireMoyen() qui affiche le salaire moyen des employés de la collection. Testez votre programme avec le main suivant :

```
class Salaires {

1       public static void main(String[] args) {

2
3       Personnel p = new Personnel();

4
5       p.ajouterEmploye(new Vendeur("Mohamed", "Salah", 45, "1995", 30000));

6
7       p.ajouterEmploye(new Representant("Ali", "Ben Ali", 25, "2001", 20000));

8
9       p.ajouterEmploye(new Technicien("Yosri", "Mansour", 28, "1998", 1000));
10

11      p.ajouterEmploye(new Manutentionnaire("Jalel", "Tounsi, 32, "1998", 45));

12
13      p.ajouterEmploye(new TechnARisque("Jamel", "Fehri", 28, "2000", 1000));

14
p.ajouterEmploye(new ManutARisque("Ali", "Abloulou", 30, "2001", 45));

p.afficherSalaires();

System.out.println("Le salaire moyen dans l'entreprise est de " + p.salaireMoyen() + "
francs.");

}
}
```