

TP03 : Polymorphisme - La classe Object - Classes abstraites

Exercice 1. Liaison dynamique

Considérons les déclarations de classes suivantes :

```
public class GestionPoint {  
    public static void main(String args[]){  
        . . .  
    }  
}  
  
class Point {  
    private int x;  
    private int y;  
    public Point(int x, int y){ . . . }  
    public String toString(){  
        return("Point [" + x + ", " + y + "]");  
    }  
}  
  
class PointColor extends Point{  
    private String couleur;  
    . . .  
    public String getCouleur(){ . . . }  
    public String toString(){ . . . }  
}
```

Compléter les définitions de ces classes (parties en pointillés). Faire de sorte que la méthode *toString()* de la classe *PointColor* affiche les informations sur le point, ainsi que la couleur.

- Supposons qu'on ait dans le corps de la méthode main les instructions suivantes :
 Point p = new Point(0, 0);
 PointColor pc = new PointColor (10, 5, "jaune");
- Quels sont les problèmes au niveau de la compilation ou de l'interprétation pour chacune des instructions suivantes ; et si tout se passe bien, analyser le résultat :
 - a. System.out.println(p);
 - b. System.out.println(pc);
 - c. p.getCouleur();
 - d. ((PointColor) p).getCouleur();
 - e. System.out.println(((Point) pc));
- Que renvoie l'instruction (new Point(10, 10)).equals(new Point(10, 10))?
- Redéfinir la méthode equals (héritée de la classe Object) de telle façon que:
 (new Point(10, 10)).equals(new Point(10, 10)) renvoie **true**.
 (new PointColor(i,j,ch1)).equals(new PointColor (i,j,ch2)) renvoie aussi **true**, avec ch1 et ch2 deux chaînes identiques au sens de equals (c'est à dire, ch1.equals(ch2) renvoie true).
- Vérifier vos propositions en faisant les tests suivants :
 Point p1 = new PointColor (0, 0, "rouge");
 Point p2 = new PointColor (0, 0, "rouge");
 System.out.println("p1 égal à p2 ? " + p1.equals(p2));
 System.out.println("p1 égal à p2 ? " + p1.equals((Object) p2));

Exercice 2. Une collection d'ustensiles

Votre grand-mère a besoin d'un programme java pour gérer sa collection d'ustensiles de cuisine anciens, précisément des assiettes (rondes ou carrées) et des cuillères. Vous trouverez ci-jointe une version incomplète de son programme `Collection`. Les cinq objets créés sont stockés dans une liste `ustensiles`.

```
package exo2;

import java.util.ArrayList;
public class Collection {
    public static final int anneCourante=2019;
    public static void main(String args[]){
        ArrayList <Ustensile> ustensiles=new ArrayList <Ustensile>();
        ustensiles.add(new AssietteRonde(1926,8.4));
        ustensiles.add(new Cuillere(1881,7.3));
        ustensiles.add(new AssietteCarree(1935,5.6));
        ustensiles.add(new Cuillere(1917,8.8));
        ustensiles.add(new AssietteRonde(1837,5.4));

        afficherUstensiles(ustensiles);

        afficherSurfaceAssiettes(ustensiles);

        afficherValeurTotale(ustensiles);
    }

    public static void afficherUstensiles(ArrayList <Ustensile> ustensiles){
        ...
    }

    public static void afficherCuilleres (ArrayList <Ustensile> ustensiles) {
        ...
    }

    public static void afficherSurfaceAssiettes(ArrayList <Ustensile> ustensiles){
        ...
    }

    public static void afficherValeurTotale(ArrayList <Ustensile> ustensiles){
        ...
    }
}
```

Il vous est demandé de compléter le programme selon les indications données 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.

Hiérarchie des classes :

Source : <https://perso.telecom-paristech.fr/hudry/coursJava/exercices/objetsCommunicants.html>



Il convient de modéliser les objets de la collection avec une hiérarchie de 5 classes comme il est indiqué dans la liste ci-dessous. Ecrivez le code de cette hiérarchie de classe, y compris les variables d'instance et les constructeurs. Vous pouvez ajouter les nouvelles classes au fichier `Collection.java` (voir dans votre répertoire partage) ou bien utiliser des fichiers séparés. La méthode `main` du programme `Collection` vous montre la façon d'appeler les constructeurs des classes qui sont instanciables.

- Chaque `Ustensile` a une année de fabrication qui est de type `int`. Un `Ustensile` est soit une `Assiette`, soit une `Cuillère`. Il n'est pas possible d'instancier la classe `Ustensile`.
- Une `Assiette` est soit une `AssietteRonde`, soit une `AssietteCarree`. Il n'est pas possible d'instancier la classe `Assiette`.
- Une `AssietteRonde` a un `Rayon` qui est de type `double`.
- Une `AssietteCarree` a un `côté` qui est de type `double`.
- Chaque `Cuillere` a une `longueur` de type `double`.

Chaque classe qui peut être instanciée redéfinira la méthode `toString` de la classe `Object`.

Affichage des ustensiles

Complétez le code de la méthode `afficherUstensiles` pour qu'elle affiche les ustensiles de la liste ustensiles.

Comptage

Complétez le code de la méthode `afficherCuillere` pour qu'elle calcule et affiche le nombre d'objets de type `Cuillere` qui sont stockés dans la liste ustensiles. Par exemple, le code de la méthode `main` 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 des assiettes stockées dans la liste ustensiles, c'est-à-dire les assiettes rondes et les assiettes carrées.

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 la liste. Il sera nécessaire de compléter la hiérarchie des classes avec des méthodes de calcul de valeur.

La valeur d'un ustensile se calcule comme suit :

- `Cuillere` et `AssietteRonde` : si l'ustensile a moins de 50 ans, il vaut 0 francs CFA. Sinon, il vaut 1000 francs CFA pour chaque année qu'il a de plus que 50 ans. Par exemple, une `AssietteRonde` ou une `Cuillere` fabriquée en 1965 vaut 4000 francs CFA.
- `AssietteCarree` : une assiette carrée vaut 5 fois la valeur qu'elle aurait eue si elle était ronde (car les assiettes carrées sont plus rares). Par exemple, une `AssietteCarree` fabriquée en 1965 vaut 5×4000 francs CFA.