

TP-6 (Java POO)

Classes Abstraites et Interfaces

Dans ce TP, nous allons travailler avec les classes et les méthodes abstraites ainsi que les interfaces.

Une grand-mère a besoin d'un programme Java pour gérer sa collection d'ustensiles de cuisine anciens, plus précisément des assiettes (rondes ou carrées) et des cuillères. Vous trouverez ci-dessous une version incomplète de son programme **MesUstensiles**. Les 5 objets créés sont stockés dans une liste **ust** qui est de type **Ustensile**.

La règle de calcul de la **valeur** par défaut d'un ustensile dépend de l'âge de l'ustensile :

Si l'âge est ≤ 50 ans, alors $valeur = 0$. Si l'âge > 50 ans, alors $valeur = age - 50$;

Donc tout ustensile ajouté à notre application, et dont on n'a pas encore de règle de calcul personnalisée, lui sera appliquée la règle par défaut ci-dessus.

Les assiettes ont une valeur particulière qui est égale à cinq fois la valeur par défaut.

Les cuillères ont une valeur particulière qui est égale à deux fois la valeur par défaut.

Il vous est demandé de compléter le programme selon les indications dans les quatre méthodes ci-après de la classe **Application** (voir page suivante).

Pensez à profiter au maximum des possibilités offertes par Java (concepts de la POO) pour éviter la duplication inutile d'instructions dans les classes et méthodes.

En exécutant la méthode **main()** de la classe **Application**, vous devriez avoir le résultat suivant :

```
Il y a 2 cuillères.
Il y a 1 assiettes carrés.
Il y a 2 assiettes rondes.
Surface totale des assiettes (carrés et rondes) : 344.6393413543701
Valeur totale de la collection : 704.0
```

```
public class Application {

    public static void main(String[] args) {

        List<Ustensile> ustensiles = new ArrayList<Ustensile>();
        ustensiles.add(new AssietteRonde(8.4f, 1926));
        ustensiles.add(new Cuillere(7.3f, 1881));
        ustensiles.add(new AssietteCarre(5.6f, 1935));
        ustensiles.add(new Cuillere(8.8f, 1917));
        ustensiles.add(new AssietteRonde(5.4f, 1837));

        afficherNombreArticles(ustensiles);
        afficherSurfaceAssiettes(ustensiles);
        afficherValeurTotale(ustensiles);
    }

    static void afficherNombreArticles(List<Ustensile> listeUstensiles) {
        int nbCuilleres = 0;
        int nbAssiettesRondes = 0;
        int nbAssiettesCarre = 0;
        for (Ustensile ust : listeUstensiles) {
            if (ust instanceof Cuillere) {
                nbCuilleres++;
            }
            if (ust instanceof AssietteCarre) {
                nbAssiettesCarre++;
            }
            if (ust instanceof AssietteRonde) {
                nbAssiettesRondes++;
            }
        }
        System.out.println("Il y a " + nbCuilleres + " cuillères.");
        System.out.println("Il y a " + nbAssiettesCarre + " assiettes carrés.");
        System.out.println("Il y a " + nbAssiettesRondes + " assiettes rondes.");
    }

    static void afficherSurfaceAssiettes(List<Ustensile> listeUstensiles) {
        double somme = 0;
        for (Ustensile ust : listeUstensiles) {
            if (ust instanceof Assiette) {
                somme += ((Assiette) ust).calculerSurface();
            }
        }
        System.out.println("Surface totale des assiettes (carrés et rondes) : " + somme);
    }

    static void afficherValeurTotale(List<Ustensile> listeUstensiles) {
        float somme = 0;
        for (Ustensile ust : listeUstensiles) {
            somme += ust.calculerValeur();
        }
        System.out.println("Valeur totale de la collection : " + somme);
    }
}
```