

TD 4
Le modèle de conception COMPOSITE

Exercice 1

Question 1/ Ecrire une classe `Met` qui possède les méthodes : `toString()` qui renvoie le nom de ce met et les informations suivantes `estSucre()`, `estSale()`, `estSucreSale()`, `nombreDeCalories()` et `estDietetique()` (nombre de kcalories inférieur à 200) .

Question 2/ Ecrire les classes `Sucre`, `Sel`, `Poire`, `Veau` qui héritent de `Met`.

Question 3/ Utilisez le patron de conception COMPOSITE pour représenter un plat composé (une poire au sucre, un veau au sel ou un veau aux poires) également comme un met particulier.

Dans ce cas, un met est sucré si tous ses éléments sont sucrés, salé si tous ses éléments sont salés et sucré-salé sinon.

Exercice 2

Cet exercice consiste à mettre en œuvre le pattern COMPOSITE par la déclaration des arbres binaires de recherche d'objets. Pour simplifier on considérera des arbres contenant des `int`.

Question 1/ Utilisez le pattern composite pour déclarer des arbres binaire de recherche. On prévoiera :

- L'insertion,
- Savoir si l'arbre est vide,
- Le test de la présence d'un élément,
- Le parcours infixe,
- Le calcul du poids de l'arbre.

Question 2/ Proposer une méthode pour supprimer une valeur dans un arbre. Plusieurs cas sont à considérer, une fois que le nœud à supprimer a été trouvé à partir de sa valeur :

- **Suppression d'une feuille** : Il suffit de l'enlever de l'arbre vu qu'elle n'a pas de fils (remplacer par un arbre vide).
- **Suppression d'un nœud avec un enfant** : Il faut le remplacer par son fils.
- **Suppression d'un nœud avec deux enfants** : On le remplace la valeur du nœud à supprimer par son successeur le plus proche (la valeur du nœud le plus à gauche du sous-arbre droit) ou son plus proche prédécesseur (la valeur du nœud le plus à droite du sous-arbre gauche). On supprime ensuite cette valeur dans le sous-arbre droit ou gauche. Cela permet de garder une structure d'arbre binaire de recherche.
