

Exercice JAVA - Objet

Exercice 1.13 : on veut réaliser un programme de gestion des recettes de cuisine, qui sera installé sur des appareils électroménagers pour leur permettre de faire la cuisine de façon autonome. Un programmeur a déjà écrit la classe Ingredient donnée ci-dessous :

```
class Ingredient{
    String nom_aliment, etat;
    int quantite;
    String unite;
    Ingredient(String n, String e, int q, String unite){
        this.nom_aliment = n;
        this.etat = e;
        this.quantite = q;
        this.unite = unite;
    }
}
```

NB: l'état d'un ingrédient peut être cuit, entier, cru, découpé, ou une combinaison de ces états (par exemple cuit et entier). L'unité peut être une unité de poids (gramme, kg, etc), de volume (litre, ml, cl) ou simplement une *cardinalité*.

a. Écrivez une classe Plat qui représente les plats, chaque plat ayant un nom et une liste d'ingrédients. On doit pouvoir créer un plat avec son nom. Il faut également avoir des accesseurs sur le nom du plat et les ingrédients, et pouvoir ajouter un ingrédient à un plat. Écrivez également une méthode main qui crée un plat appelé *choucroute* contenant comme ingrédients : 500g de choucroute cuite, 150g de lard cuit et entier et 2 saucisses entières et cuites

b. On veut pouvoir comparer les plats et donc leurs ingrédients. Ajoutez une méthode equals dans la classe Ingredient qui renvoie true si deux ingrédients ont le même nom d'aliment et le même état (pas forcément la même quantité). Ajoutez une méthode equals dans la classe Plat, qui renvoie true si deux plats contiennent les mêmes ingrédients, au sens donné juste avant.

c. On veut faire la distinction entre les ingrédients qu'on peut cuire et ceux qu'on peut découper. Un ingrédient qu'on peut cuire doit avoir une méthode cuire() qui le fait passer dans l'état "cuit" et une température de cuisson. Un ingrédient qu'on peut découper doit avoir une méthode decouper() qui le fait passer dans l'état "découpé". Proposez du code objet pour représenter ces types d'ingrédients.

Exercice 3.9 : l'Office National des Forêts (ONF) veut réaliser un programme pour gérer les données récoltées par les ingénieurs des eaux et forêts concernant les arbres et les parcelles. Un début de code a été écrit par un programmeur qui ne connaît pas la POO. Proposez un nouveau programme conforme aux principes de la POO et qui offre les mêmes possibilités de représentation des données et les mêmes fonctionnalités.

```
public class ONF_Data {
    private int typeArbre; // 0:résineux, 1:feuillu
    private int hauteur, diametre;
    private int parcelle; // chaque parcelle est identifiée par un entier
    private int[] surfaces; // surfaces[i] est la surface de la parcelle numéro i
    public static void creerArbre(int type, int h, int d, int parcelle) {
        ...
    }
    // affecte une surface à une parcelle
    public static void setSurface(int parcelle, int surface) {
        ...
    }
    // retourne le nombre d'arbres sur une parcelle
    public static int nombreArbres(int parcelle) {
        ...
    }
}
```

Exercice 2.11 : on veut écrire un programme pour simuler le comportement des élus. Un programmeur a écrit la classe ci-dessous (avant de démissionner pour se présenter aux élections législatives).

```
class Personne{
    String nom, prenom;
    int compteBancaire; // montant de son compte en euros
    Personne(String nom, String prenom){
        this.nom = nom; this.prenom = prenom;
        this.compteBancaire = 0;
    }
    void addSous(int montant){
        this.compteBancaire = this.compteBancaire+montant;
    }
}
```

En réutilisant cette classe Personne, écrivez une classe qui représente les élus, qui sont des personnes avec un ensemble d'assistants (qui sont aussi des personnes bien sur). Un élu peut embaucher un assistant ou le licencier. Un élu peut aussi distribuer à ses assistants une dotation budgétaire : il répartit la somme qui lui est allouée de façon équitable entre ses assistants en ajoutant des sous sur les comptes bancaires des assistants.

a. Écrivez une classe Elu qui représente les élus.

Dans la suite, on veut représenter différents types d'élus, mais qu'on veut pouvoir manipuler comme des élus "normaux", c'est-à-dire qu'un objet de type Elu n'aura toujours que les trois méthodes définies dans la classe Elu et permettant d'embaucher un assistant, de licencier un assistant et de verser une dotation.

b. On veut représenter les élus mafieux, qui n'embauchent comme assistants que des membres de leur famille (c'est-à-dire des personnes qui ont le même nom qu'eux, pour simplifier). Mais ils répartissent la dotation budgétaire qui leur est allouée de façon équitable entre leurs assistants. Écrivez une classe qui représente les élus mafieux.

c. On veut représenter les élus économes, qui répartissent la dotation budgétaire qui leur est allouée de la façon suivante : chaque assistant reçoit au plus le SMIC (1480 euros brut) et s'il reste des sous, ils ne sont pas dépensés. Écrivez une classe qui représente les élus économes.

d. On veut représenter les élus escrocs, qui ont un compte en Suisse en plus de leur compte bancaire normal, qui n'embauchent pas forcément que les membres de leur famille, et qui répartissent la dotation budgétaire qui leur est allouée de la façon suivante : chaque assistant reçoit au plus le SMIC (1480 euros brut) et s'il reste des sous, ils sont versés sur le compte en Suisse de l'élus. Écrivez une classe qui représente les élus escrocs.