CENTRE DE RÉADAPTATION MULHOUSE

Rééducation et Formation Professionnelle

Exercice POO

sous-titre

CONTENU

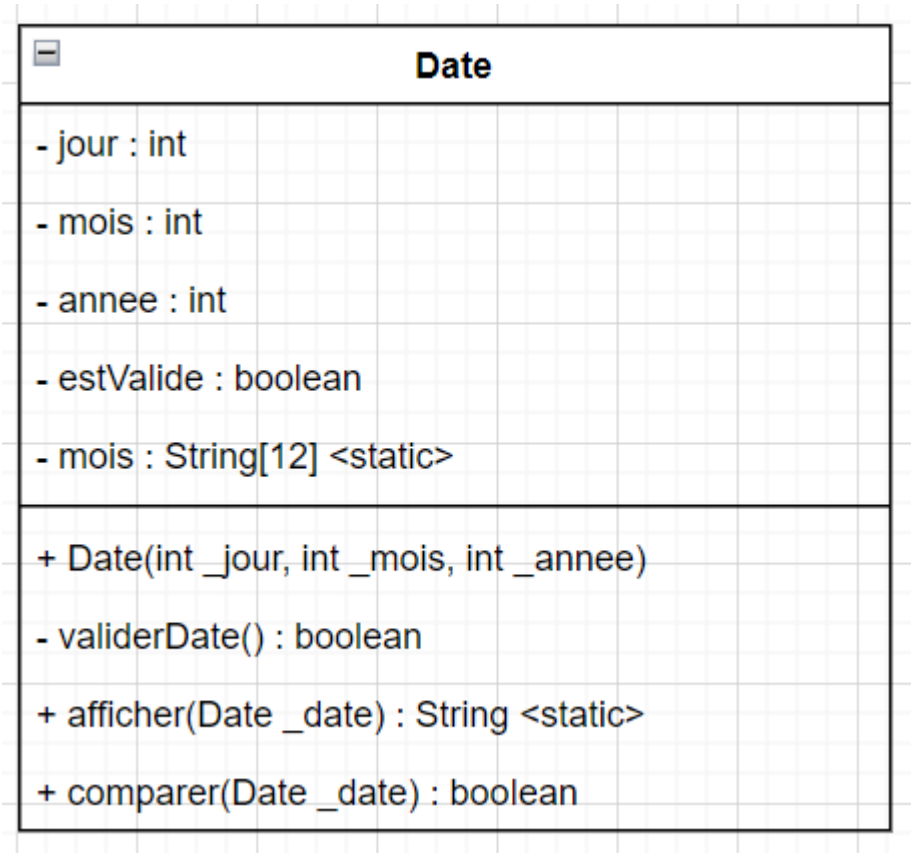
Exercice Date 1

Exercice IMC 2

Exercice Recette de cuisine 3

EXERCICE DATE

Soit la classe Date définie par le diagramme UML suivant :



Ecrire cette classe en Java.

La méthode validerDate() , vérifie la validité d'une date, passe à true le boolean estValide si la date est une date valide (par exemple le 32/14 /2020 n'est pas une date valide) et renvoie true ou false.

Rappel : Une année est bissextile, si elle est divisible par 4 mais pas par 100, ou bien si elle est divisible par 400.

La méthode de classe afficher(Date _date) permet d'afficher la date sous la forme 01 février 2024, les noms des mois seront définis en tant qu'attribut de classe à l'aide d'un tableau.

La méthode comparer(Date _date) permet de comparer 2 objets Date. L'expression d1 < d2 doit grâce à cette méthode renvoyer true ou false(d1 et d2 étant deux objet Date).

EXERCICE IMC

Réaliser le diagramme de classe de la classe Personne ayant 3 attributs définissant certaines caractéristiques d'une personne réelle : taille, poids et âge.

Cette classe aura :

- Une méthode imc() permettant de calculer l'IMC d'une personne.
- Une méthode interpretation() qui affiche le résultat selon le tableau suivant :

IMC	Interprétation OMS	Risque de maladie
Moins de 16	Maigreur extrême	Elevé
16 à 18,4	Insuffisance pondérale	Moyen
18,5 à 24,9	Corpulence normale	Faible
25 à 29,9	Surpoids	Moyen
30 à 34,9	Obésité	Elevé
35 à 40	Obésité sévère	Très élevé
Plus de 40	Obésité morbide	Extrêmement élevé

Rappel : L'IMC (indice de masse corporel) est calculé par la formule poids/taille² avec le poids en kg et la taille en m.

Coder cette classe en Java, et coder la classe App permettant de tester votre classe IMC.

EXERCICE RECETTE DE CUISINE

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

```java
package RecetteCuisine;

public class Ingredient {

        private String nomAliment, etat;
        private int quantite;
        private String unite;

    public Ingredient(String _nomAliment, String _etat, int _quantite, String _unite)
    {
        this.nomAliment = _nomAliment;
        this.etat = _etat;
        this.quantite = _quantite;
        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).

a. Écrire 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. Écrire 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 comparer dans la classe Ingrédient qui renvoie true si deux ingrédients ont le même nom d'aliment et le même état (pas forcément la même quantité). Ajoutez une méthode comparer 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.

--- FIN DU DOCUMENT ---