

Dans tout ce TD, on précisera pour chaque méthode la classe dans laquelle elle est écrite.

1. On souhaite implémenter une structure de liste chaînée pour modéliser des colliers de nouilles colorées. En plus de la classe `Couleur` vue en cours, on envisage une classe `Nouille` avec un attribut `val` de type `Couleur` et un attribut `suiivante` de type `Nouille` et une classe `Collier` avec un attribut `debut` de type `Nouille`. **Tous les attributs ci-dessus sont privés.**
 - (a) Écrivez ces deux classes en ajoutant,
 - dans la classe `Collier`, un constructeur qui crée un collier vide
 - dans la classe `Nouille`, deux constructeurs : un qui prend un argument de type `Couleur` et un argument de type `Nouille` et un constructeur qui prend seulement un argument de type `Couleur`.
 - (b) Écrivez une méthode `ajouter` qui prend en argument une couleur et crée et ajoute une nouille de cette couleur au début du collier.
 - (c) Écrivez le code permettant d’instancier un collier de quatre nouilles de couleur tomate, nature, épinard et nature respectivement. On supposera que dans la classe `Couleur`, on a déclaré les attributs `public final static Couleur TOMATE, EPINARD et NATURE` en les initialisant à des valeurs adéquates.
2. On ajoute dans les classes `Collier` et `Nouille` les méthodes suivantes :

```

// dans la classe Collier
public int mystere(Couleur c){
    if (debut == null)
        return 0;
    else
        return debut.mystere(c);
}

//dans la classe Nouille
public int mystere(Couleur c){
    int n;
    if (val.egalA(c))
        n = 1;
    else
        n = 0;
    if (suiivante == null)
        return n;
    else
        return n + suiivante.mystere(c);
}
  
```

- (a) Soit `col` le collier créé dans la question 1.(c), que feront les appels suivants écrits dans un `main` d’une classe `Test`.

```
System.out.println(col.mystere(Couleur.NATURE));  
System.out.println(col.mystere(Couleur.EPINARD));
```

Plus généralement, que calcule la méthode `mystere` de `Collier` ?

- (b) Donnez une méthode itérative qui calcule la même chose que `mystere`.
3. Écrivez en récursif et en itératif les méthodes suivantes
- (a) `griser` qui grise chaque nouille du collier. On suppose que dans `Couleur`, on a la méthode `public Couleur grisEq()` qui retourne une couleur grise de même intensité lumineuse que `this`.
 - (b) `afficher` qui affiche les couleurs des nouilles du collier dans l'ordre.
4. Écrivez en récursif uniquement, une méthode `afficherInv` qui affiche les couleur des nouilles dans l'ordre inverse.
5. Écrivez en récursif et en itératif la méthode `dernier` qui retourne la dernière nouille du collier.

Facultatif

7. Écrivez en récursif et en itératif les méthodes suivantes :
- (a) une méthode `existe(Couleur c)` qui retourne `true` si et seulement si il y a au moins une nouille de la couleur donnée en argument.
 - (b) Écrivez en récursif et en itératif une méthode `toutes(Couleur c)` qui retourne `true` si et seulement si toutes les nouilles sont de la couleur donnée en argument.