# TD-TP : Relation bidirectionnelle **1xN** en Java

## MISE EN PLACE

Un `PlatCuisiné` est caractérisé par un `libelle` et une `recette`, tous les deux du type `String`. Chaque `PlatCuisiné` est composé de plusieurs `Ingredient`, chaque `Ingredient` étant caractérisé par un `libellé`. Un `Ingrédient` ne pouvant participer qu'à un seul `PlatCuisiné`

### Travail à faire

**1.** Proposer le schéma de classe UML modélisant cette situation

**2.** Créer un projet Java sous Eclipse, intitulé 3.`RelationSymetrique1xN` dans lequel coder les classes `Ingrédient` et `PlatCuisiné`.

**3.** Coder un `main()` dans une classe `TesterRelationSymetrique1xN` qui crée un `ingrédient1<pain>`,
un `ingrédient2<beurre>`, et un `ingrédient2<fromage>`, puis qui affiche la valeur de ces objets via `toString()`.

# IMPLEMENTATION DE LA RELATION BIDIRECTIONNELLE 1xN

Il s'agit maintenant l'implémenter la relation bidirectionnelle **1xN**.

## Travail à faire

**4.** Enrichir le schéma de classe de la réponse **1.** pour intégrer la double navigabilité entre `Ingrédient` et `PlatCuisiné`.

**5.** Modifier le code des classes `Ingrédient` et `PlatCuisiné` de sorte à intégrer les attributs, et méthodes correspondantes, sachant que :

**a.** Classe `Ingrédient` :
- pour se délier un plat « Si l'ingrédient est lié à un plat, le retirer de la liste de ses ingrédients, et ne plus pointer vers ce plat. »
- pour se lier à un nouveau plat « Le délier du plat avec lequel il est éventuellement lié, pointer vers le nouveau plat, et faire pointer symétriquement le nouveau plat vers moi ».

**b.** Classe `PlatCuisiné` :
- pour ajouter, retirer et dire si un ingrédient existe, il faut utiliser les méthodes de `ArrayList` (cf. Parcourir un `ArrayList` sur eLearn).
- pour se lier à un ingrédient, « S'il n'est pas présent dans la liste, l'ajouter, puis supprimer son lien éventuel et le faire pointer vers moi »
- pour se délier d'un ingrédient « Le supprimer de la liste et l'ingrédient ne doit plus pointer vers moi ».

**c.** Afin d'expérimenter les « TroisFaconsDeParcourirUnArrayList » présentées sur eLearn, écrire trois méthodes `toString()` chacune dédiée à une des façons de parcourir un ArrayList. `toString1()` parcours l'ArrayList avec un accès direct aux éléments, `toString2()` qui fait un parcours séquentiel de l'ensemble des éléments de l'ArrayList et `toString3()` qui parcours l'ArrayList avec un itérateur.

**6.** Dans le `main`, tester votre solution :

**a.** Créer deux plats cuisinés `plat1 <tartineBeurée>` et `plat2 <fromageSeul>`, lier `plat1` avec `ingrédient1` et `ingrédient2`, puis lier `plat2` avec `ingrédient3`. Finir en affichant `plat1` et `plat2` avec `toString`()

**b.** Créer un plat cuisiné `plat3 <sandwichAuFromage>`, lier `plat3` avec `ingrédient1`, `ingrédient2` et `ingredient3`, puis afficher `plat1`, `plat2` et `plat3` avec `toString()`. **Constater que `plat1` et `plat2` ont été démunis de leurs ingrédients...**