TD 1 - Introduction à la Programmation Orientée Objet

TD d’application POO : Classes et encapsulation

# Bienvenue dans le monde merveilleux de la POO !

## Introduction

Les choses sérieuses commencent… Pour mettre en application le cours, je vous propose un TD qui va vous accompagner à la mise en pratique de ce que l’on a vu en cours précédemment. Pour cela, nous allons nous intéresser au robot de cuisine.

Nous allons, au fur et à mesure des cours, venir compléter ce TD et appliquer les principes de la POO.A

## Contexte

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. Celui-ci aura la possibilité d’enregistrer des plats et, surtout, les ingrédients qui composent les plats. Nous allons nous intéresser tout d’abord aux ingrédients. Pour chaque ingrédient, nous allons nous intéresser à leur nom, la quantité nécessaire ainsi que leur unité. L'unité peut être une unité de poids (gramme, kg, etc), de volume (litre, ml, cl) ou simplement une cardinalité.

# Prérequis

* Connaitre les bases de JAVA
* Savoir créer un projet
* Savoir poser des questions (n’est-ce pas ?)

# Ressources utiles

* La procédure de création d’un projet
* La procédure d’import de JDBC
* Le 1er cours sur la POO

Durant le TD, n’hésitez pas à nous solliciter afin de valider certaines étapes si vous avez un doute !

# Début du TD

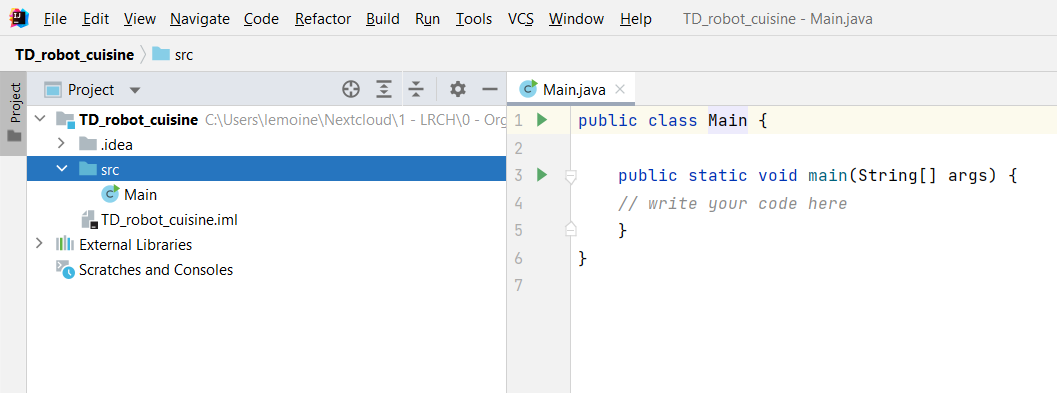
## Etape 1 : Création du projet

Prêt ? C’est parti !

* 1re étape, vous allez créer un projet nommé « TD\_robot\_cuisine » que vous placez ou vous le souhaitez. Pensez bien que nous allons faire évoluer ce projet donc mettez-le à un endroit où vous êtes sûr de pouvoir le retrouver au fur et à mesure des séances. Sinon, vous pouvez faire en sorte de l’exporter en Zip pour le mettre sur votre OneDrive. Ou bien, il est possible aussi de le mettre sur GitHub ! (On verra ça plus tard…)
  + Si vous avez un doute, vous pouvez vous rendre sur OneDrive pour trouver la procédure de création d’un projet

Travail à faire : Créer le projet nommé « TD\_robot\_cuisine »

* Si tout s’est bien passé et que vous avez suivi le tutoriel de création d’un projet, vous devez avoir ceci à droite dans la section projet :



* On doit y retrouver notre classe « Main » contenant notre fonction de lancement « main() ».
  + Nous allons garder ce fichier pour être notre porte d’entrée vers notre application de cuisine.
  + Si vous ne l’avez pas, vous pouvez créer ce fichier, on en aura besoin ensuite.

## Etape 2 : Création de notre classe « Ingrédient »

* Créons notre classe Ingrédient en mettant bien la Majuscule à Ingrédient
  + La Majuscule fait partie des standards de la Programmation Orientée Objet. Cela nous permet notamment de distinguer les variables ou attributs des objets.

Travail à faire : Créer la classe « Ingrédient » dans le dossier « src »

## Etape 3 : Création des attributs et méthodes

* L’objectif de cette classe va être de nous permettre :
  + De renvoyer une chaine de caractère avec les attributs sous le format suivant : « ‘nom de l’ingrédient’ : ‘quantité’ ‘unité’ »
    - Exemple : « Beurre : 20 gr »
    - Ail : 1 gousse
    - Sel : 1 c.à.s
  + De nous permettre de récupérer le nom de l’ingrédient, la quantité, l’unité
  + De modifier (seulement) la quantité de l’ingrédient si nécessaire
* Nous allons donc commencer par la création des attributs de notre classe car nous en connaissons une partie grâce au sujet. Nous avons donc :
  + L’attribut « nom » de type String
  + L’attribut « quantite » de type int
  + L’attribut « unité » de type String
* Pour respecter le principe d’encapsulation, chaque attribut doit être déclaré en private
  + De plus une un attribut, comme une variable, doit commencer par une minuscule

Travail à faire : Créer les trois attributs de votre classe

* Concernant notre classe, nous avons besoin de créer plusieurs méthodes (4 en tout)
  + La méthode afficher :
    - Elle ne prend pas de paramètre
    - Elle doit simplement retourner la chaine de caractère suivante : « ‘nom de l’ingrédient’ : ‘quantité’ ‘unité’ »
  + La méthode getNom :
    - Elle ne prend pas de paramètre
    - Elle doit simplement retourner la valeur de l’attribut nom
  + La méthode getQuantite :
    - Elle ne prend pas de paramètre
    - Elle doit simplement retourner la valeur de l’attribut quantité
  + La méthode getUnité :
    - Elle ne prend pas de paramètre
    - Elle doit simplement retourner la valeur de l’attribut unité
  + La méthode setQuantite :
    - Elle doit prendre un paramètre qui est la nouvelle quantité
    - Elle doit affecter la nouvelle quantité à notre attribut quantité de notre classe

Travail à faire : Créer les quatre méthodes de votre classe

Vous pouvez nous appeler pour valider le tout.

## Etape 4 : Création du constructeur

* Nous avons implémenté les méthodes, créer les attributs, nous sommes prêts à interagir avec nos futurs objets de type « Ingrédient »
* Mais, il nous reste un problème, comment je crée mon ingrédient ?
  + Par défaut chaque objet à un constructeur vide qui permet de l’instancier même sans avoir créer un constructeur, ceci nous permet de faire :
  + Ingredient monIngredient = new Ingredient() ;
  + Or, notre problème ici, c’est que l’on a besoin de venir renseigner le nom, l’unité et la quantité de notre ingrédient. C’est pour cela que l’on va écrire notre propre constructeur pour donner les informations nécessaires à la création de notre « Ingredient ».
* La création de notre constructeur aura donc besoin de 3 paramètres
* Un constructeur s’écrit comme ceci : Visibilité nomConstructeur (paramètre)
  + Il n’y a pas le type de retour comme dans une \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Travail à faire : Créer le constructeur notre classe Ingredient

## Etape finale : création d’une petite application de gestion d’ingrédient !

* Notre classe est prête pour un essai. Nous allons, dans notre classe « Main » venir compléter la fonction « main() » pour créer et essayer notre application :
* Voici le déroulé souhaiter de notre programme de test :
  + Demander à l’utilisateur de saisir le nom de l’ingrédient
  + Demander à l’utilisateur de saisir la quantité de l’ingrédient
  + Demander à l’utilisateur de saisir l’unité de l’ingrédient
  + Instancier un objet de type « Ingredient » dans une variable nommée « monIngredient » grâce à votre constructeur et les valeurs saisi par l’utilisateur.
  + Afficher ensuite seulement la quantité de votre ingrédient grâce à la méthode getQuantite()
  + Afficher ensuite seulement le nom de votre ingrédient grâce à la méthode getNom()
  + Afficher ensuite seulement l’unité de votre ingrédient grâce à la méthode getUnite()
  + Afficher ensuite l’ingredient en entier grâce à sa fonction afficher.
  + Demander à l’utilisateur de saisir une nouvelle quantité
  + Modifier la quantité de votre ingrédient grâce à la méthode setQuantite([…])
  + Re-afficher ensuite l’ingrédient en entier grâce à sa fonction afficher.
    - Est-ce que la quantité a bien changé ?

## Etape complémentaire

Pour ceux qui ont terminé ou ceux qu’ils souhaitent s’entrainer à la maison

* Créer un nouveau fichier nommé « MainBonus » afin de ne pas se mélanger avec le 1er
* A l’intérieur de ce nouveau fichier, créez un nouveau main()
* L’objectif ici est de créer une petite application permettant à un utilisateur de venir créer plusieurs ingrédients que l’on viendra enregistrer dans une ArrayList.
* Cette application aura donc un menu avec :
  + La possibilité de créer un nouvel ingrédient et de le placer dans notre ArrayList
  + La possibilité d’afficher l’ensemble des ingrédients présent dans notre ArrayList
  + La possibilité d’afficher un seul ingrédient de notre liste à la position donné.
  + La possibilité de venir remplacer un ingrédient par un autre créé juste avant
  + La possibilité de supprimer un ingrédient de notre ArrayList
  + La possibilité de modifier la quantité d’un ingrédient à la position donné.