

TP – Programmation Orientée Objet

Licence 2 Informatique

Bienvenue à tous dans les concours de cuisines Tok Chef.fe de l'INSA, certifiés Hauts-de-France. Pour leurs mises en place, l'institut décide de faire appel à ses meilleurs développeurs (non, non ne vous retournez pas, on parle bien de vous) afin de créer la librairie d'objets nécessaire à leur gestion. Le langage support choisi est Java (c'est plutôt une chance comme c'est le langage qui est utilisé pour la mise en pratique du cours !). Voilà les informations et les données fonctionnelles qui vous permettrons de concevoir et de réaliser les objets nécessaires. Tout Easter Egg est le bienvenu cru, mollet, dur au plat ou à la coque.

- Pour l'organisation de la section cuisine, l'institut gère trois catégories de personnes :
 - Les chef.fes : qui sont identifiés par un id, leur nom, prénom, téléphone, genre, nombre d'étoiles, spécialité (cuisinier, pâtissier, travail à distance, etc.). Chaque chef.fe sait réaliser un certain nombre de plats (sinon forcément.... ils ne seraient pas chef.fes !)
 - Les padawans : qui sont identifiés par un id, leur nom, prénom, genre, téléphone. Ils sont forcément sous la coupe d'un chef.fe et d'un seul. Par contre, rien n'empêche un chef.fe d'exploiter un nombre illimité de padawans.
 - Les membres de jurys : qui sont identifiés par leur nom, prénom, id, genre. On peut savoir à combien de jurys ils ont participé (ça permet de bizuter un membre du jury qui intervient pour la première fois dans un concours). Un membre de jury peut être prévu sur plusieurs jurys à venir.
- Pour faire des plats, il faut des ingrédients (c'est mieux ! Passer chez Ronald n'est pas recevable à un concours de cuisine). Les ingrédients ont un identifiant et un nom. On les catégorise dans quatre groupes : les épices (poivre, paprika, curry, safran, etc.), les viandes/poissons, les légumes, et les "autres" (dans "autres" vous pouvez mettre ce qui reste, ou, si vous avez le temps, raffiner avec d'autres catégories. Le but de cette limitation est de simplifier le travail à réaliser)
 - Les épices : n'apportent pas de calories, peuvent être bio ou non
 - Les viandes/poissons : apportent des calories et ont un taux de graisse et de gras (fonction du poids). Peuvent être bio.
 - Les légumes : apportent des calories et ont un taux de fibre (fonction du poids). Peuvent être bio ou non.
 - Les autres : apportent au moins des calories (fonction du poids), et peuvent être bio ou non. Pour le reste vous pouvez adapter en fonction des instances que vous allez créer et de l'architecture que vous avez conçue.
- Un plat est composé d'un ensemble d'ingrédients, chaque ingrédient étant préparé pour ce plat selon des consignes (cru, cuit, découpé, entier, etc.) et avec une certaine quantité. La quantité pour les épices est gérée en pincées, pas au poids (il y a toujours des exceptions partout...). Un plat peut se voir ajouter une quantité de sel (en pincées). L'avantage tout de même, et c'est la magie de Tok Chef.fe, c'est qu'il n'est pas nécessaire de gérer les différentes opérations et incorporations de la recette dans un certains ordre, le plat est réussi dans tous les cas !
- Et enfin les concours ! Un concours nécessite 3 membres de jury et au minimum 5 chef.fes qui participent. Plusieurs concours peuvent être prévus à différentes date,

mais dans ce cas ,tant que le prochain chronologiquement parlant n'est pas terminé, il n'est pas possible de commencer les autres. Chaque chef.fe présente un plat au concours (forcément un plat qu'il sait réaliser, sinon → conseil de discipline). Un concours se termine quand tous les plats des chef.fes participant au concours ont été notés. Dans ce cas, les informations du concours sont conservées pour la postérité. Par contre il se passe des choses pour le grand gagnant du concours ! En effet le nombre d'étoiles d'un chef.fe dépend de son nombre de victoires à un concours. Victoires > 1 = 1 étoile, Victoires > 3 = 2 étoiles, Victoires > 5 = 3 étoiles. Au moins dans ce monde humaniste, ils ne peuvent pas perdre d'étoiles ! Du plus quand un chef.fe gagne un concours, son padawan le plus ancien devient chef.fe (il faut penser à la relève).

Réalisation :

Vous devez proposer une solution pour gérer les données et les fonctionnalités demandées. Les principaux points à adresser sont :

- Pouvoir ajouter des protagonistes, des ingrédients, des plats, des concours etc.
- Automatiser la gestion des fins de concours.
- Organiser les concours et la liste des concours à venir.
- Évaluer des plats.
- Lister/afficher les différentes informations : concours prévus, listes de chef.fes, de padawans, des plats, d'ingrédients, etc. (de toutes façon ça vous sera utile pour les tests). Pour rappel les `System.out.println()` ne sont tolérés que dans un main ou éventuellement des tests unitaires !
- Il est possible d'afficher tous les concours auquel un chef.fe est inscrit.e.
- Il est possible d'avoir l'affichage de tous les concours auquel un chef a participé.
- On peut afficher la liste des plats connu d'un chef.fe par nombre de calorie.
- On peut aussi afficher les plat 100% bio.
- Il est possible de connaître le classement provisoire d'un concours (quand tous les plats n'ont pas été notés)

Les diverses actions seront testées dans un programme principal et/ou dans des tests JUnit (dans ce cas en prévoir plus que les 3 demandés minimum pour illustrer les possibilités de votre réalisation). Vous pouvez utiliser des Random si besoin pour éviter la saisie de trop d'informations pour les cas où c'est possible (calories, taux, etc.). Dans tous les cas essayez de faciliter la partie test, expérimentation et saisie du programme !

Utilisez au maximum ce qui a été vu en cours (sans excès, par exemple on ne fait pas de l'héritage pour le plaisir mais parce qu'il y a un intérêt) :

- Interfaces, héritage, listes, etc.
- **Et les bonnes pratiques !!**

Contraintes :

- Tous vos choix et réalisations doivent rester logiques (il ne peut pas y avoir 3 fois le même chef.fe dans un concours, pas de plats à base uniquement d'épices, pas de padawan dirigé par deux chef.fes, un padawan ne peut pas s'inscrire à un concours, etc, etc.)

- Vous devez proposer au moins trois tests JUnit sur des éléments que vous choisirez. Un des trois utilisera des données provenant d'un fichier texte facilement éditable pour son jeu de test)
- Il doit être possible de sauvegarder et de charger dans et depuis un fichier toutes les données de la section cuisine. (Ca permettra également de faciliter vos tests)
- Il doit être possible d'afficher la liste de tous les ingrédients connus en JSON.
- Un ingrédient peut être sauvegardé dans une base de données. Juste montrer que vous savez persister un objet (voir exemple TD). Vous pouvez mettre en place cette fonctionnalité dans un projet à part si vous le souhaitez, pour faciliter la mise en œuvre.

Méthode d'évaluation :

Le TP donnera lieu à une note sur 20 qui prendra en compte l'architecture et la conception de l'application à base d'objets, la clarté du code, et les objectifs atteints. Avoir une application fonctionnelle mais sans objet ou très mal conçue est à éviter. Ce n'est pas que "ce qui marche" qui est évalué, mais surtout "comment ce qui fonctionne est conçu"

Conseils :

- Concevez ce que vous allez développer AVANT de commencer à coder, cela évitera d'avoir à perdre du temps avec une architecture bancale et devoir tout recommencer.
- Ne définissez pas un grand nombre d'ingrédients ou de plats, définissez-en assez pour réaliser les expérimentations nécessaires.