

UNIVERSITÉ DE SHERBROOKE  
Faculté de génie  
Département de génie électrique et génie informatique

## **RAPPORT D'APP**

Modèles de conception  
APP 1

Présenté à  
Domingo Palao Munoz

Présenté par  
Équipe numéro 7  
Paul du Réau de la Gaignonnière - durp0701  
Olivier Lortie - loro0801

Sherbrooke – 9 septembre 2020

# **TABLE DES MATIÈRES**

<b>1.</b>	<b>Diagramme de classe</b>	<b>1</b>
<b>2.</b>	<b>Utilisation des modèles de conceptions</b>	<b>1</b>
<b>3.</b>	<b>Questions</b>	<b>3</b>
<b>Annexe A</b>	<b>Diagramme de classe</b>	<b>4</b>

# 1. DIAGRAMME DE CLASSE

Voir l'ANNEXE A : Diagramme de classe ou dans le dossier comportant le code.

## 2. UTILISATION DES MODÈLES DE CONCEPTIONS

Tableau 2-1 : Utilisation des modèles de conception

Modèle de conception	Utilisation	Justification
State	Gérer les états d'une facture	Ce modèle a été utilisé pour pouvoir gérer les changements d'états d'une facture, sans avoir à gérer une série de conditions dans le code de la facture elle-même. De plus, si un état est ajouté plus tard, il n'y aura pas de code à modifier.
State	Gérer les états des plats choisis	Ce modèle a été utilisé pour pouvoir gérer les changements d'états d'un plat choisi, sans avoir à gérer une série de conditions dans le code du plat lui-même. De plus, si un état est ajouté plus tard, il n'y aura pas de code à modifier.
Observer	Pour notifier un chef lorsqu'un plat choisi est ajouté à une facture	Le but est de permettre à un ou plusieurs chefs de savoir lorsqu'un plat est ajouté à une facture pour pouvoir le préparer. De cette façon, le chef n'a pas besoin de toujours vérifier si un plat a été ajouté à une facture.
Factory Method	Pour créer des plats au menu différents (plats enfants et plats santé)	Le but de ce modèle de conception est de permettre de créer des plats au menu différents (plats enfants ou plats santé) sans nécessairement comprendre ce qui se passe dans ces sous-classes. En effet, on peut les créer à partir d'une même interface, qui est notre factory.

**Tableau 2-2 : Utilisation des modèles de conception (suite)**

<b>Modèle de conception</b>	<b>Utilisation</b>	<b>Justification</b>
Adapter	Adapter les ingrédients liquides pour les rendre solides	Au départ, les ingrédients liquides ne sont pas compatibles avec les ingrédients solides, car ils ont une quantité en ml au lieu d'être en g. On ajoute donc un adapter pour transformer la quantité en g et que l'ingrédient soit compatible avec les autres.
Adapter	Adapter les plats enfants pour qu'ils aient la bonne quantité d'ingrédients selon leur proportion	Au départ, le plat enfant n'est pas compatible avec les plats au menu puisqu'il contient seulement une portion des ingrédients du plat. On doit donc lui ajouter un adapter pur qu'il puisse être compatible avec tous les autres plats.
Iterator	Itérer à travers les ingrédients de l'inventaire	Le but est de pouvoir itérer à travers les ingrédients de l'inventaire, sans nécessairement connaître la structure des ingrédients dans l'inventaire.
Iterator	Itérer à travers les ingrédients d'un plat au menu	Le but est de pouvoir itérer à travers les ingrédients d'un plat, sans nécessairement connaître la structure des ingrédients du plat.
Singleton	Permet d'avoir un seul et unique inventaire	Il doit pouvoir exister un inventaire seulement. Dans un restaurant, il n'y a pas plusieurs inventaires

### 3. QUESTIONS

Question 1 : Quel est l'avantage d'avoir utilisé des Modèles de Conception pour développer l'application ? Est-ce que vous recommandez cette manière de créer une application ?

Réponse : Les Modèles de Conception sont des méthodes connues pour résoudre des problèmes communs. Ils respectent aussi les cinq principes «SOLID» qui permettent d'implémenter un code qui possède une bonne architecture facile à comprendre, flexible, facilement extensible et adaptable. Oui nous recommandons cette architecture. Dans la majorité des cas elle permet de simplifier le code, sa compréhension ainsi que son extension, mais les Modèles de Conception sont très spécifiques à une situation donnée, ce qui limite les possibilités de trouver le modèle adéquat. Par exemple, dans notre utilisation du modèle de conception Factory Method, nous n'avons pas l'impression que c'était vraiment nécessaire de l'utiliser, puisque nous ne faisons que bouger le constructeur dans une autre classe. On croit donc que son utilisation n'était pas tout à fait optimale dans cette situation particulière.

Question 2 : Quel est l'avantage d'avoir utilisé les Tests Unitaires dans le développement de l'application ? Est-ce que les Tests Unitaires ont changé votre manière de développer l'application ?

Réponse : Les Tests Unitaires permettent de vérifier le bon fonctionnement de chaque fonction du code de manière indépendante afin de déterminer la fonction qui pourrait ne pas fonctionner. Les Tests Unitaires ont changé notre façon de programmer. Après avoir implémenté chacune des fonctions, nous faisons ces tests pour vérifier qu'elles fonctionnaient correctement avant de faire la suivante. Par exemple, ces tests nous permettaient de vérifier si notre adapter sur les ingrédients fonctionnait correctement avant de pouvoir ajouter les ingrédients aux plats.

## Annexe A      DIAGRAMME DE CLASSE

