

# **COMPTE RENDU TP C++ : TP3**

***Rédigé et réalisé par :***

NZOKOU Ives Chatelin & SABI Abdellah

***Année académique :***

2022-2023

# INTRODUCTION

Le but de ce TP3 était de créer un programme en C++ de magasin en ligne nommé Easy store en abordant ainsi les concepts de la programmation orienté objet (POO) tel que : **l'encapsulation, l'héritage, le polymorphisme, abstraction**. Nous avons donc relevé quelques objets (entités) qui vont interagir dans notre application et crée un certain nombre de classe de classe les représentants. Une classe est la représentation d'une entité du monde réel caractérisé par ses états et se comportements.

## 1. Pour quoi le choix de ce TP ?

Nous avons choisi ce TP de niveau intermédiaire parce que un d'entre nous à un niveau débutant et l'autre un niveau intermédiaire en programmation orienté objet et ce TP et est adapté pour les débutant et intermédiaires. Cela nous permettra de consolider nos bases et acquis théoriques et de revoir tous les notions dispensées en cours.

## 2. Répartition du travail

Nous avons réparti le travail de selon les compétences de chacun comme suite :

Abdellah (niveau débutant) : création des classes Client et Commande et toutes les fonctions et méthodes contenue dans celles-ci.

Nzokou (Niveau intermediaire) : création des classes Magasin et Produit et toute les fonctions et méthode contenue dans celles-ci. La création du programme main.

## 3. Outils utilisés

Nous avons utilisé un outil de travail collaboratif : **replit**, que nous avons associé à un compte github.

## 4. Développement du code

### a. Définition des classes

Nous avons défini les classes qui vont interagir dans notre programme de gestion de réservation d'hôtel. Nous avons ainsi identifié dans l'énoncé ce TP, 4 Classes qui seront manipulés. Le tableau ci-dessus représente les classes créés et les fonctions principales mis à part les getters et les setters et quelques méthodes secondaires sont :

Classe	Variable	Type	Méthode/fonction
	_produits	vecteur	addClientMagasin

Magasin	_clients	vecteur	getAllClientMagasin getClient
	_commandes	vecteur	addProduitPanierClient deleteProduitPanierClient updateQteProduitClient validerCommande updateStatusCommande getAllCommande
Client	_identifiant	string	Surcharge de l'opérateur
	_nom	string	<<
	_prenom _paniers	String vecteur	addProduitpanier videPanier modifierquantite supprimerProduit
Produit	_title	string	updateQuantite
	_description	string	isValid
	_quantite _prix	int double	operator<<
Commande	_numCom	int	updateStatus
	_idclient	string	Operator<<
	_produits _status	vecteur Enum	

## b. Travail réalisé

Nous avons implémenté toute les fonctions du programme et ajouter quelques petits apports à certaines classes :

- Nous avons utilisé une Enumération pour définir le statut de la commande.
- Pour les fonctions de la classe Magasin nous les avons définis comme des fonctions amies.
- Nous avons utilisé les étiquettes dans le **main** pour accéder à différent menu avec l'option retour dans l'application.

- L'application créer les fichier commandes.txt, client.txt et produit.txt s'ils n'existent pas et s'ils existent il rajoute l'objet à la suite du fichier.
- Nous avons géré la récupération la saisi de chaine de caractère avec des espaces pour le prénom et la description des produits.
- Nous avons ajouté une variable identifiant à la classe client.
- Un identifiant de commande est créé aléatoirement entre 0 et 100.
- La création de chaque objet est stockée dans son fichier mais pas les modifications de celles. Car fait juste la lecture et écriture des fichiers.
- La liste des produits et des commandes est lue à partir de ses fichiers.
- Les toutes les fonctions du programme sont liées aux 3 tableaux produit client commande. Il faut en rajouter les valeurs pour effectuer mes modifications sur eux.
- L'application que le makefile crée, est appelé **store**.

## CONCLUSION

Après avoir réalisé toutes les taches qui ont été demandé sur ce TP et avoir ajouté quelques codes propres à nous. Nous avons touché presque toute les notions de la programmation orienté objet. Nous avons beaucoup appris sur les fonctions amies, le polymorphisme et l'encapsulation. Nous avons encore beaucoup à apprendre et nous devons programmer plus pour avoir un niveau plus avancé qui nous sera très utilise dans notre parcourt professionnelle plus tard.