

TP 17 : Gestion des commandes

L'objet de cette application est la gestion informatisée des commandes de clients, Elle doit permettre de mettre à jour la quantité stock de chaque produit commandé.

1. Classe **Client**

- a. Définir une classe Client dont les caractéristiques sont : Code client, Nom, Adresse, Tél, Email.
- b. Ecrire un constructeur à deux paramètres obligatoires : Code client, Nom.
- c. Ecrire la méthode `__str__()` qui renverra le Nom et l'adresse du client .

2. Classe **Produit**

- a. Définir une classe Produit dont les caractéristiques sont : Références, désignation, Prix unitaire, Quantité Stock.
- b. Ecrire la méthode `__str__()` qui renverra les informations de ce produit mais sans la quantité en stock.

3. Classe **Commande**

- a. Définir une classe Commande dont les caractéristiques sont :
 - Numéro de commande : le numéro de la première commande créer est 1001 et à chaque création d'une nouvelle commande, ce numéro doit être incrémenté de 1.
 - Cclient : c'est le client qui a passé la commande
 - Date commande : date où Cclient a passé cette commande
 - Une liste des produits commandés et une liste des quantités commandées : ces deux listes doivent toujours avoir le même nombre d'éléments : à chaque produit commandé correspond une quantité commandée dans la deuxième liste.
- b. Ecrire un constructeur sans paramètres qui permet de définir la valeur du numéro de commande et d'affecter la date système à la date commande
- c. Ecrire la méthode `__str__()` qui renverra le numéro de commande, la date commande et le nom client séparés par tabulation.
- d. Ecrire la méthode **TotalCommande()** qui retourne le coût global de la commande =Somme des (Prix unitaire * Quantité commandée) pour l'instance en cours.
- e. Ecrire la méthode **AjouterProduit()** ayant comme paramètres le produit à commander et la quantité commandée correspondante: Elle doit permettre la mise à jour de la liste des produits commandés, la liste des quantités commandées et la quantité stock de ce produit ajouté (Nouvelle quantité stock = Ancienne quantité stock - quantité commandée).
- f. Ecrire une méthode **AfficherCommande()** qui permet d'imprimer sur l'écran les détails d'une commande de la manière suivante :

Date :/../....			
N° Commande :			
Nom client :			
Adresse :			
Réf.	Désignation	Prix unitaire	Qté. Commandée
Montant			
.....

Indication : Montant est une donnée calculée = Prix unitaire * Qté. Commandée

4. Programme principal

- a. Déclarer une liste de clients, une liste de produits et une liste de commandes accessibles par toutes les méthodes du programme principal.
- b. Ajouter une méthode **rechercherClient()** qui recherche un client par son code. Si trouvé, elle retourne le client correspondant dans la liste des clients. Sinon, elle retourne la valeur None.
- c. Ajouter une méthode **rechercherProduit()** qui recherche un produit par sa désignation, Si trouvé, elle retourne le produit correspondant dans la liste des produits. Sinon, elle retourne la valeur None.
- d. Ajouter une méthode **rechercherCommande()** qui recherche une commande par son numéro. Si trouvé, elle retourne l'indice correspondant dans la liste des commandes. Sinon elle retourne la valeur -1.
- e. Ajouter par code (en mode conception) trois clients dans la liste des clients.
- f. Ajouter par code trois produits dans la liste des produits.
- g. Ajouter par code deux commandes dans la liste des commandes. La 1^{ère} contenant un produit et les 2^{èmes} deux produits. Utiliser certains constructeurs et certaines méthodes déjà conçus.

Concevoir le menu suivant : (en tenant compte des indications ci-dessous)

1- Ajouter client.

2- Ajouter produit.

3- Ajouter commande.

4- Rechercher commande par numéro commande.

5- Rechercher les commandes passées à une date.

6- Fin.

Indications concernant les traitements à réaliser dans le menu ci-dessus :

- Ajouter commande :
 - ✓ Le numéro de la nouvelle commande est automatiquement affecté par programme et la date commande correspond à la date système.
 - ✓ Saisir le code du client qui doit correspondre à un client de la liste des clients et si trouvé, le client ainsi trouvé correspond à Cclient de la commande à ajouter, sinon le programme vous demandera de saisir un autre code client.
 - ✓ Pour chaque produit à commander, l'utilisateur saisira sa désignation à chercher dans la liste des produits (même principe que celui du code client). Si trouvée, le produit ainsi trouvé sera ajouté à la liste des produits de la commande en cours.
 - ✓ Chaque fois que la désignation saisie est trouvée, saisir la quantité commandée correspondante au produit trouvé. Cette dernière sera soustraite de la quantité stock de ce produit.
 - ✓ Après la saisie de la quantité commandée, le programme demandera à l'utilisateur s'il souhaite saisir un autre produit pour la commande en cours.

- Rechercher commande par numéro commande : saisir le numéro de commande. Si trouvé, imprimer la commande comme précisé dans la figure en haut (méthode `AfficherCommande()`). Sinon, afficher un message adéquat pour avertir l'utilisateur de l'inexistence de cette commande.
- Rechercher les commandes passées à une date : Imprimer la liste des commandes (Numéro commande, Date commande, Nom client) dont la date commande est égale à la date saisie. Si aucune commande ne correspond à cette date saisie, afficher un message adéquat.