

Projet 11 : Gestion des stocks

NB :

- **Toutes les fonctionnalités de ce projet doit être réaliser en utilisant la bibliothèque stl. (pour parcourir, utiliser les itérateurs, pour les traitements, utiliser les algorithmes)**
- **Pour la date, utiliser une structure pour stocker le jour, le mois et la date.**

On considère la conception suivantes :

- **Classe Dépôt:**
 - Attributs privés : id dépôt, adresse, capacité ;
 - Relations : Dépôt – Stock (composition 1 à *)
- **Classe Stock:**
 - Attributs privés : référence stock, description stock .
 - Relations : Stock – Produit (1 à *), Stock – Dépôt (* à 1) ;
- **Classe Produit :**
 - Attributs privés : référence produit, désignation, quantité, prixHT ;
 - Relations : classe de base de la classe Produit Electronique, Produit – Stock (* à 1), Produit- Fournisseur (* à *);
- **Classe Produit électronique :**
 - Attributs privés : version du matériel, version du logiciel .
 - Relations : sous classe de la classe Produit;
- **Classe Fournisseur :**
 - Attributs privés : id fournisseur, nom, contact ;
 - Relations : Fournisseur – Produit (* à *), Fournisseur – Paiement(1 à *)
- **Classe Paiement :**
 - Attributs privés : id paiement, montant, date paiement ;
 - Relations : Paiement – Fournisseur (* à 1), Paiement – Produit (* à *)
 - Méthode : fiche paiement

NB : la fiche paiement doit contenir les informations suivantes :

- Identifiant paiement :
- Identifiant fournisseur :
- Liste des produits :
- Date paiement:

Référence produit	Désignation	Quantité	Prix
....
....

- Total à payer :

Travail à faire :

1. Créer les différentes classes ;
 2. Dans le main
 - a. Créer un menu de navigation pour les différentes fonctionnalités
- **Fonctionnalité 1 : gestion Stock ****
- a. Dans un `std::set` demander à l'utilisateur de remplir les stocks.
 - b. Afficher le nombre des produits de chaque stock
 - c. Permettre à l'utilisateur d'ajouter/modifier/supprimer un stock
- **Fonctionnalité 2 : gestion Fournisseur ****
- a. Dans un `std::deque` demander à l'utilisateur de remplir les fournisseurs
 - b. Permettre à l'utilisateur supprimer un fournisseur donné
 - c. Afficher les produits dont le prix est inférieur à 100dh

****Fonctionnalité 3 : gestion des Produits ****

NB : utiliser les informations déjà enregistrées dans le `std::deque` et le `std::set`

- a. Dans un `std::map`, associer la référence produit avec un objet produit et demander à l'utilisateur de remplir les produits ;
- b. Trier la liste des fournisseurs de chaque produit par son prix
- c. Permettre à l'utilisateur de ajouter/modifier/supprimer un fournisseur
- d. Afficher le nombre des produits dont la date de paiement à dépasser les 2 mois
- e. Réduire le prix du produit de 10% si la quantité est supérieure à 200 pièces

****Fonctionnalité 4 : gestion Paiement****

NB : utiliser les informations déjà enregistrées dans le `std::set`, `std::deque` et le `std::map`

- a. Dans un `std::list`, demander à l'utilisateur de remplir la liste des paiements,
- b. Ajouter 15.5% de TVA sur les produits de chaque paiements
- c. Permettre à l'utilisateur de modifier le fournisseur d'un paiement donnée
- d. Permettre à l'utilisateur de ajouter /supprimer un produit d'un paiement donnée
- e. Afficher pour chaque paiement la fiche de paiement