

### TP 3 : Gestion Magasin

---

<b>Objectifs :</b>	<ul style="list-style-type: none"><li>▪ Notion de classe et d'objet</li><li>▪ Déclaration de classe</li><li>▪ Déclaration des attributs et des méthodes</li><li>▪ Les types des variables (primitifs et objets)</li><li>▪ Notion de référence</li><li>▪ Les constructeurs</li></ul>
--------------------	---

#### Exercice 1:

La boîte de développement « TuniProd » désire développer une application de gestion de ressources de magasin de produits alimentaires.

Avant d'entamer la réalisation de ce programme, la société a commencé par une étude du marché qui a permis de dégager les différentes caractéristiques d'un produit alimentaire et qui sont :

- ☐ Un identifiant (entier)
- ☐ Un libellé (chaîne de caractère)
- ☐ Une marque (chaîne de caractère)
- ☐ Un prix (un nombre réel)

Le programme doit permettre à l'utilisateur de :

1. Créer un produit vide (sans attributs)
2. Créer des nouveaux produits tout en spécifiant les caractéristiques suivantes :
  - a. 1021, Lait et Delice

b. 2510, Yaourt et Vitalait

c. 3250, Tomate, Sicam et 1.200

3. Afficher les détails de chaque produit créé en implémentant une méthode `afficher()`.
4. Attribuer le prix 0.700 au produit lait, afficher le produit modifié.
5. Compléter les informations manquantes pour chaque produit
6. Afficher les produits modifiés.
7. Afficher les produits déjà créés avec la méthode `toString()`. Qu'est-ce que vous avez remarqué ?
8. Ajouter un attribut « date d'expiration » de type `Date` et affecter des dates aux produits existants.

## **Exercice 2 :**

Dans le cadre de l'amélioration du programme sur lequel vous êtes affectés dans l'entreprise

« TuniProd » et dans un souci constant d'adaptation aux besoins du client, vous êtes demandés de créer un autre module afin de gérer les besoins des propriétaires des magasins.

Les différentes caractéristiques d'un magasin:

- ☐ Un identifiant
  - ☐ Une adresse
  - ☐ Capacité du magasin
  - ☐ Un ensemble de produits
1. Le module doit permettre aux propriétaires des magasins en premier lieu d'ajouter un nouveau produit au magasin, tout en prenant en considération qu'un magasin peut contenir au maximum 50 produits.
  2. Vous êtes aussi demandés d'afficher les caractéristiques du magasin, et le nom et le prix de l'ensemble ses produits.

3. Afin de protéger les différentes caractéristiques des objets créés votre supérieur vous demande de restreindre l'accès aux attributs de la classe Produit en prenant en considération que le prix de ce dernier ne doit pas être négatif.
4. Ensuite il vous suggère d'organiser le code en des différents packages avec le nom commençant par « tn.tuniprob.gestionmagasin ».
5. Finalement le programme doit permettre à son propriétaire de savoir le nombre total de tous les produits de tous les magasins.

### **Exercice 3 :**

Une fois le client a testé les fonctionnalités précédemment développées (ajout et l'affichage des produits dans les magasins), il constate quelques problèmes et lacunes au niveau de la gestion des produits, tels que la possibilité d'avoir des redondances de certains produits dans le même magasin.

Pour cela votre supérieur vous demande de faire les modifications nécessaires afin de remédier à ce problème.

### **Travail demandé**

- 1) Créer une méthode « **comparer** » qui permet de tester la conformité de produits, sachant que le client considère deux produits identiques dans le cas où ils ont le même identifiant, libellé et prix. Ceci doit être fait avec deux façons :
  - a. méthode « comparer » prend un seul produit en paramètre.
  - b. méthode « comparer » prend deux produits en paramètre.
- 2) Ajouter une autre méthode qui sert à chercher un produit dans un magasin, la méthode retourne « int » dans le cas où le produit est trouvé.
- 3) Modifier la méthode « **ajouterProduit** » afin d'ajouter un produit donné une seule fois
- 4) Créer une méthode qui permet de supprimer un produit du magasin.
- 5) Créer une méthode qui affiche le magasin ayant un nombre supérieur de produits entre deux magasins.