

TD-TP : *Produit-Magasin-Panier-Achat*

Si vous le souhaitez, les classes `Produit`, `Magasin` et `Panier` peuvent être définies les unes sous les autres dans l'unique fichier `main.cpp`

Considérant les déclarations :

```
typedef string  Libelle;
typedef string  Code;
typedef float   Prix;
typedef int     Quantite;
typedef string  Adresse;
```

il s'agit de produire des sections de code relatives à des produits, des magasins et des paniers.

Les Produits

- A. Donner la déclaration de la classe `Produit` dont une instance est définie par trois attributs `libelle`, `code` et `prix` de types respectifs `Libelle`, `Code` et `Prix`. Donnez la signature (cad version `.hpp`) des constructeurs que vous jugez utiles.
- B. Donner le code de la méthode `string Produit::toString()`
- C. Créer dans le `main()` les trois produits suivants `p1=<"TABLE", "T001", 110>`, `p2=<"CHAISE", "C001", 50>` et `p3=<"FAUTEUIL", "F001", 70>` et afficher leur valeur

Les Magasins

On définit une classe `Magasin` dont une instance mémorise pour le magasin correspondant, son stock de produits ainsi que l'adresse du magasin. Pour cela l'attribut `stock` du magasin est du type `map` dont la clef est un `Produit*` et la valeur est du type `Quantite`.

- D. Donner les lignes de code correspondant à la déclaration de la classe `Magasin` (version `.h`)
- E. Ecrire la méthode `void Magasin::toString()` qui affiche l'adresse du magasin de même que son `stock`, à savoir chaque produit en stock et quantité correspondante.
- F. Ecrire dans le `main` les instructions relatives à la création d'un magasin `monMagasin` localisé à Anglet qui stocke une table `T001`, deux chaises `C001` et 3 fauteuils `F001`, puis qui édite les valeurs de `monMagasin`

Les Paniers

On définit une classe `Panier` dont une instance mémorise pour le panier correspondant, les achats et le magasin concerné. Pour cela l'attribut `achats` du panier est du type `list` dont les éléments sont des `pair` du type `Produit*` acheté et `Quantite` achetée. L'attribut `magasin` est du type `Magasin*`

- G. Donner les lignes de code correspondant à la déclaration de cette classe `Panier`
- H. Ecrire le code de la méthode `string Panier::toString()` qui affiche l'adresse du magasin ou le panier a été constitué, la liste des achats présents dans le panier et le prix/cout total du panier.

L'achat

- I. Ecrire le code de la méthode :

```
bool Panier::acheter(Produit* produitEnvisage, Quantite quantiteEnvisagee)
```

qui met à jour le panier de même que le stock du magasin selon que le `produitEnvisage` est présent en `quantiteEnvisagee`.

- J. Créer un panier dans le magasin d'Anglet, essaye d'acheter 2 tables `T001`, 1 chaise `C001` et 1 fauteuil `F001` puis afficher le panier constitué de même que les données du magasin `monMagasin`.